1

DEVELOPMENT OF AN AUTOMATIC GROUND COLLISION AVOIDANCE SYSTEM USING A DIGITAL TERRAIN DATABASE

THESIS

Gregory W. Bice
Captain, USAF

AFIT/GAE/ENY/89D-03

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

(1)

AFIT/GAE/ENY/89D-03

DEVELOPMENT OF AN AUTOMATIC GROUND COLLISION AVOIDANCE SYSTEM USING A DIGITAL TERRAIN DATABASE

THESIS

Gregory W. Bice
Captain, USAF

AFIT/GAE/ENY/89D-03

AFIT/GAE/ENY/89D-03

DEVELOPMENT OF AN AUTOMATED GROUND COLLISION AVOIDANCE SYSTEM USING A DIGITAL TERRAIN DATABASE

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Aeronautical Engineering

Gregory W. Bice, B.S.

Captain, USAF

December 1989

Preface

The purpose of this study was to develop a working control system that would perform automatic ground collision avoidance using a digital terrain database. A secondary purpose was to show the potential of the digital terrain database for improving the mission capabilities of combat aircraft. Both of those purposes were fulfilled in this thesis.

The topic studied in this thesis has current applications to the Air Force, therefore, I feel work should continue to be devoted to this area of research. Potential savings in both aircraft and pilots make automated ground collison avoidance a worthwhile endeavor.

In developing and writing this thesis, my thanks and appreciation go to many people who have made the rough road a little smoother. I am very thankful for the engineering prowess and persistance of my thesis advisor, Capt Curt Mracek. His understanding and assistance made the hard times in this thesis a little easier. Thanks also go to Capt Brett Ridgely for his assistance in control system analysis. I also wish to extend a hand of appreciation to my sponsor Mr. Finley Barfield of the Flight Dynamics Laboratory for the use of facilities, assistance in deciphering control law diagrams, and his expert knowledge of the F-16. Under the area of morale, I wish to thank all of my friends in the Bullpen for their humor and support. I will miss the gatherings of the "Friday at the Flywright" gang who helped make AFIT a bearable place. Finally, I am eternally thankful for the support of my wife, Susan, who put up with my late nights, bad days, and gave me a wonderful daughter, Lauren. Thanks Lord.

Gregory W. Bice

<u>Table of Contents</u>

Page

List of Figures

Figure | Page

Figure		Page

List of Tables

ABSTRACT

During the past several years, the Air Force has experienced an increasing number of single seat aircraft mishaps due to what is termed 'controlled flight into terrain'. To combat this phenomenon, several ground collision avoidance systems (GCAS) have been developed to warn the pilot of a potential collision with the terrain if some action is not taken. However, all current systems have shortcomings pertaining to the sensors that are used and the recovery maneuver that is flown. The USAF is evaluating the potential of digital terrain databases for onboard navigation and terrain avoidance in combat aircraft. The purpose of this thesis was to develop a control system for performing terrain avoidance using a simulated terrain database. This study was conducted for an F-16 aircraft in level flight at 0.6 Mach and sea level conditions. A state space model of the aircraft and its flight control system was developed using aircraft control derivatives, an F-16 control law diagram, and traditional linearization techniques on the aircraft equations of motion. A control system for implementing terrain avoidance was derived based on the look-ahead capability of the terrain database. Control system response was evaluated using a simulated terrain obstacle and various look-ahead distances on the terrain database. Results indicated that a 1200 foot or roughly 1.8 second look-ahead distance provided good improvement in terrain avoidance capabilities for the F-16 compared to looking strictly downward from the aircraft for terrain information.

DEVELOPMENT OF AN AUTOMATED GROUND COLLISION AVOIDANCE SYSTEM USING A DIGITAL TERRAIN DATABASE

I. Introduction

Background

During the past four to five years, the Air Force has recognized that an increasing number of accidents in fighter and attack aircraft, such as the F-16 and A-10, have been due to a phenomenon called 'controlled flight into terrain', or CFIT. These are accidents in which good aircraft, flown by capable pilots, crash into the terrain due to pilot incapacitation, disorientation, or distraction. Aggressive maneuvers performed at low altitude, such as breaking off of the target after weapon release, can cause g-induced loss of consciousness GLOC) and spatial disorientation; the latter happening more at night or in clouds where reference points can become lost. The rise in the number of CFIT accidents can in part be attributed to the increased emphasis that has been placed on the close air support / battlefield air interdiction (CAS/BAI) role.

To combat the problems presented by CFIT, several systems have been developed to help in preventing CFIT accidents. These systems, called ground collision avoidance systems (CGAS) or ground proximity warning systems (GPWS), monitor aircraft states such as altitude above ground level (AGL), airspeed, and attitude. This information is in turn fed to a computer algorithm which calculates a pull-up initiation altitude that will allow the aircraft to avoid impacting the terrain or penetrating a pre-determined buffer altitude. Whenever the pull-up altitude is equal to or less than the actual AGL altitude of the aircraft, a warning is sent to the pilot that he must initiate a prescribed pull-up maneuver. One system, developed for use on the Advanced Fighter

Technology Integration (AFTI)/F-16, performed the pull-up maneuver automatically by rolling to a wings-level attitude and performing a 5-g pull-up (1:21). This automated capability, while having several advantages over the previously described manual GCAS systems, has not been put into operational use due to computer and autopilot limitations.

Current GCAS Limitations. While these GCAS implementations have worked to varying degrees by saving pilots and aircraft, they have limitations. First is the issue of manual versus automated recovery. A manual GCAS must incorporate an allowance for pilot reaction time into its pull-up calculations, and, since reaction times vary from pilot to pilot, the pull-up maneuver will not be identical. Furthermore, this type of GCAS relies solely on the pilot to recover the aircraft once a pull-up warning is given; pilot incapacitation breaks the recovery system loop. The automated GCAS recovery maneuver has the capability to be highly repeatable and consistent because it is not reliant on the pilot, hence, the allowance for pilot reaction time is not necessary. The disadvantages of an automated GCAS are the computer limitations of current aircraft and pilot distrust of automated recovery systems (1:41). Reference 1 examines the issue of pilot-vehicle interface in greater detail.

The second limitation in all current GCAS schemes lies in the sensors that feed terrain information into the collision avoidance algorithm. Radar altimeters are currently used to provide this data, however, they essentially look downward from the aircraft and have limited look-ahead capability. This is a major drawback when traversing over rough to semi-rough terrain which tends to render a GCAS useless. Aircraft possessing forward-looking radars, such as the B-1B and the F-111, implement terrain following systems which are related to ground collision avoidance systems in a broad sense; the difference being a GCAS should operate as a backup system while the pilot or autopilot is flying the aircraft. Most fighter and attack aircraft do not possess large

forward-looking radars and must rely on a radar altimeter for terrain information, however, advances in the area of digital terrain databases may solve this problem.

Digital Terrain Database. The digital terrain database (DTD) has the capability to store large areas of terrain in compact form such as a cassette tape and uses an inertial navigation unit to update aircraft location. Using a DTD will give onboard systems the ability to analyze terrain 360 degrees around the aircraft, eliminate the requirement for a forward sensor, and greatly enhance covert capabilities. With the DTD, future GCAS systems will be able to perform 'smarter' pull-up recovery maneuvers by having the capability to maneuver over and around the terrain obstacle, not merely pulling up to avoid it (1:39-41). This will provide the pilot with a safety system that will not degrade mission performance. The question that must be addressed then is how the terrain avoidance system should be implemented and what should it accomplish aside from avoiding the terrain.

Problem Statement

This study will attempt to derive a recovery maneuver based on the capabilities of the digital terrain database to 'see' terrain ahead of the aircraft. The idea behind this approach to the terrain avoidance problem is to provide the aircraft with maneuvering capabilities so that it can continue on a pre-planned mission course while also avoiding threatening terrain. Because of the importance of being at a specified set of conditions during ingress to the target area, the terrain avoidance system should also return the aircraft to its initial conditions before the recovery maneuver was initiated. All solutions and results will be predicated on the assumption of perfect terrain data correlation and registration. A linear state-space representation of the aircraft and control system will be constructed so that computer programs such as $MATRIX_X$ can be used to analyze aircraft responses (Reference 7). Inputs consisting of pitch rate and roll rate will be made to the control system through the autopilot control paths. The theory for the

basis of the recovery maneuver will be derived, and terrain avoidance capabilities will be evaluated for several different look-ahead distances on the DTD. Finally, the results of this study will be examined and conclusions drawn as to what the minimum required look-ahead distance might be. Recommendations will be made for further study and development of the terrain avoidance problem.

II. State Space Model Development

Methodology

In order to facilitate the development of a ground collision avoidance system, a state space model of the F-16 was created. Research showed that a model for the design conditions of M = 0.6 and sea level altitude did not exist, and, therefore, one had to be developed using the control derivatives for the F-16. The trim conditions and control derivatives for this condition are detailed in Appendix A. Appendix B contains a layout of the F-16 along with angular definitions, and sign conventions for control surface deflections.

In order to construct a state-space representation of any control system, a condition must be selected about which to linearize the equations of motion. The control law diagram, which is not shown, was linearized about the conditions of M = 0.6 and an altitude of sea level. No pilot inputs were used, and therefore, all paths associated with pilot inputs can be ignored, as can all trim inputs. Furthermore, since the horizontal tail is normally used to command both pitch and roll rates, an effective aileron/flaperon input was created so that the longitudinal axis motions could be decoupled from those of the lateral-directional axis. This effective aileron deflection was defined to be the flaperon deflection plus one-fourth of the horizontal tail deflection:

$$\delta_{F_{eff}} = \delta_F + .25\,\delta_{HT} \tag{2.1}$$

where:

$\delta_{F_{eff}}$ = effective flaperon deflection (deg)

δ_F = flaperon deflection (deg)

δ_{HT} = horizontal tail deflection (deg)

This effective flaperon deflection was used only for roll rate commands; there was no aileron deflection when the horizontal tail was used to command normal load factor. The values of the control derivatives were also adjusted using the same formula as Eq (2.1).

The only other modification made to the control law diagram was changing the longitudinal autopilot from commanding load factor to commanding pitch rate. This involved adding several gains to convert the commanded pitch rate to normal load factor using the steady-state Z-axis acceleration equation:

$$A_n = qU_o / [(57.3)(32.2)] \tag{2.2}$$

where,

A_n = normal acceleration at pilot station (g)

q = pitch rate (deg/s)

U_o = steady-state forward velocity (ft/s)

Figures 2.1 and 2.2 show the final configuration of the linearized F-16 control laws which are separated into the longitudinal axis and lateral-directional axis respectively. The control laws have been put into a more conventional form to aid in visualizing the feedback paths.

Matrix Development

A state-space system was use to facilitate analysis of aircraft response. This involved selecting a Mach number and altitude about which the equations of motion would be linearized. The selected conditions are listed in Table 2.1.

Table 2.1: Selected Trim Conditions for Linearized Model

Mach = 0.6	Altitude = sea level
True Airspeed (V_T) = 670 ft/s	Pressure (P_a) = 2116.216 lb/ft^2
Impact Pressure (q_c) = 583 lb/ft^2	(q_c/P_a) = 0.2755

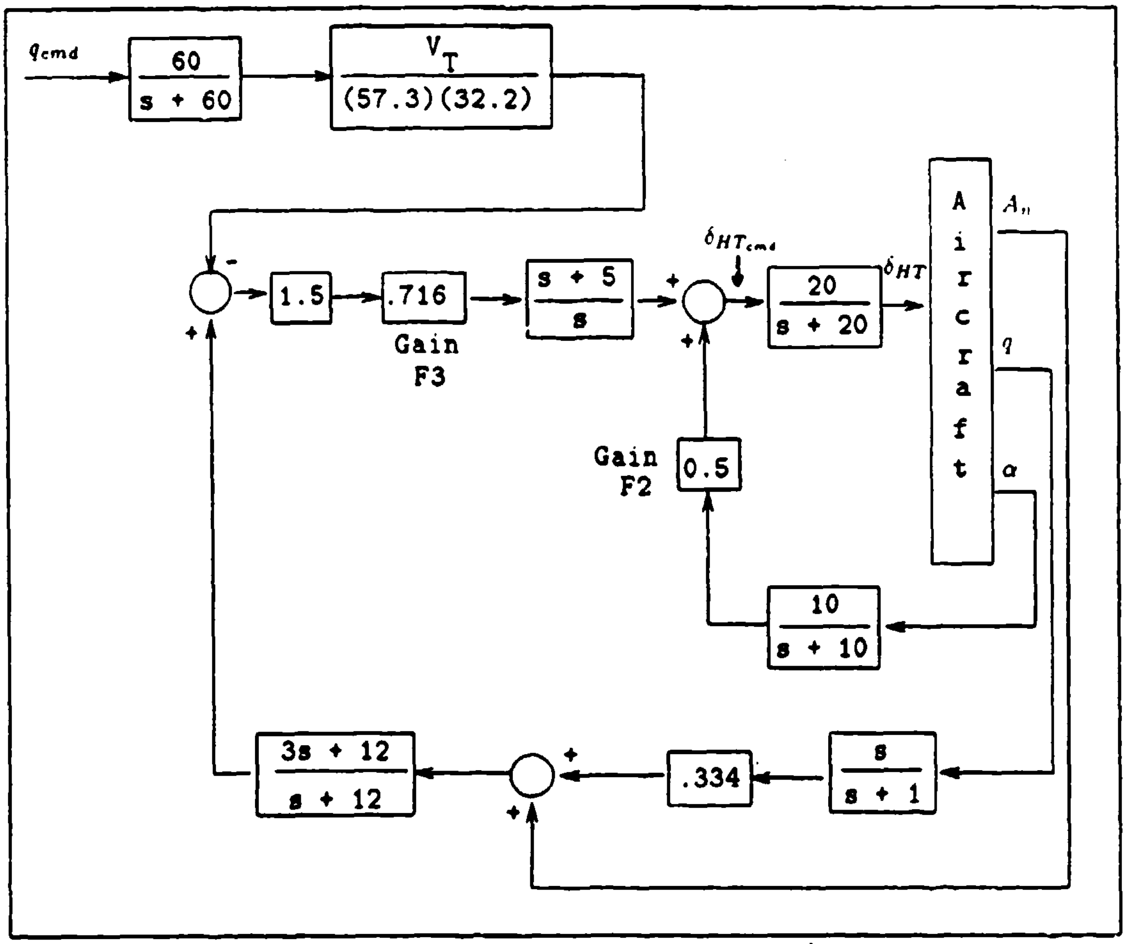


Figure 2.1: Modified F-16 Longitudinal Control System

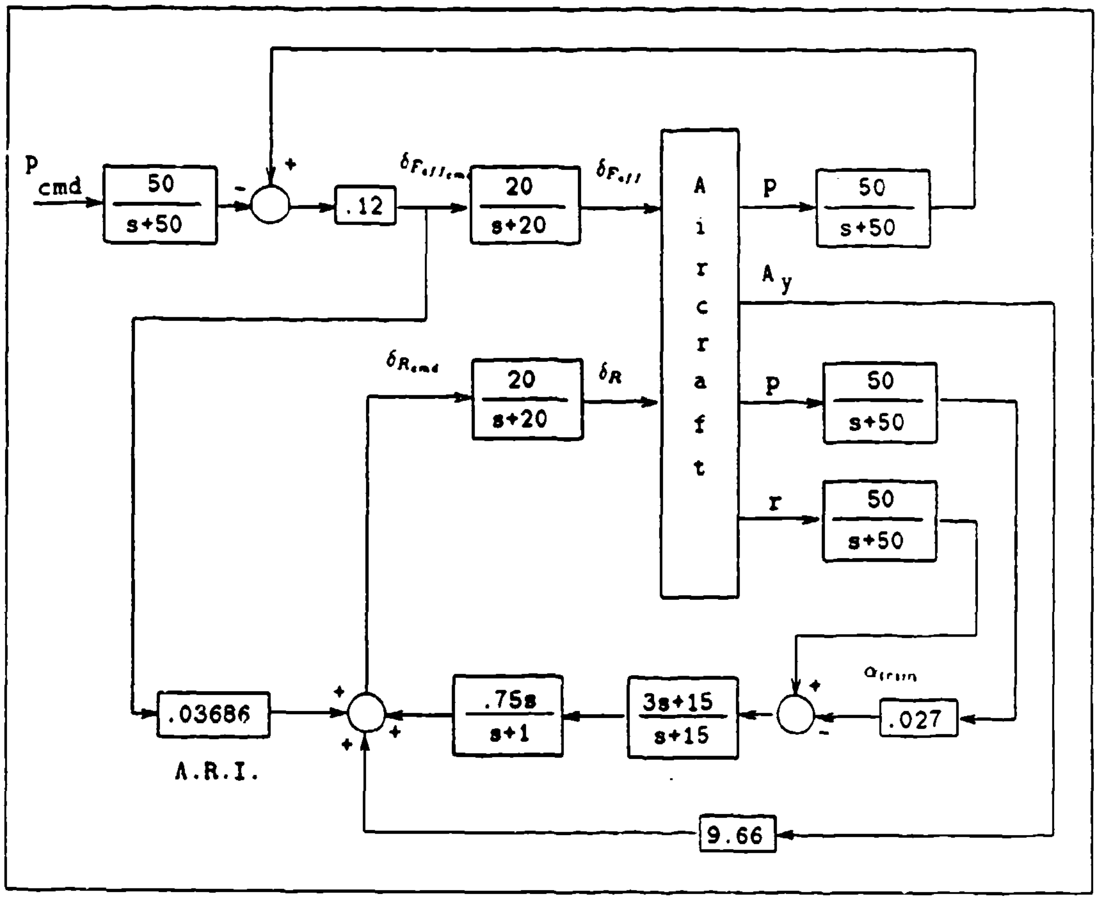


Figure 2.2: F-16 Lateral-Directional Control System

It is important to note that impact pressure, q_C, is not the same as dynamic pressure, $q = (0.5)\rho V^2$. The reason for noting this is that the scheduled gains for the control system are based on impact pressure and not dynamic pressure. There were several reasons for selecting the listed conditions, first being the fact that this is situated well within the envelope of the F-16. A second reason was that by selecting sea level conditions, any potential mistakes with pressure and density ratios are avoided since these ratios are normally used to calculate true airspeed, static pressure, and impact pressure at altitude. The final rationale for selecting these conditions was the requirement for a 5-g load factor capability without incurring very high angles-of-attack which would violate the small angle approximations made during the linearization process.

Data on the control derivatives were obtained from the Flight Dynamics Laboratory (WRDC/FIGX) for the stated conditions. Values for the control derivatives were given in the stability axis, and a computer program, listed in Appendix C, was used to convert these values to the aircraft body axis (8:276). Appendix D details the development of the equations of motion and the control derivatives and their placement in the state-space matrix (8:236). The equations of motion were developed using perturbation techniques and ignoring all terms that were second order and higher. For purposes of convenience, the system state-space matrix was broken down into the longitudinal and lateral-directional axes to aid in forming the closed loop system. This could be done since these two axes were decoupled from each other. The closed loop derivation of each axis will now be addressed separately.

Longitudinal Axis. The states used in building the longitudinal state-space system were

$$\underline{X} = [\ u \quad \alpha \quad \theta \quad q \quad \delta_{HT} \quad h_{mil}\ ]^T$$

where

$$u = \text{incremental forward velocity (ft/s)}$$

$$\alpha = \text{perturbation angle of attack (deg)}$$

$$\theta = \text{pitch angle (deg)}$$

$$q = \text{pitch rate (deg/s)}$$

$$\delta_{HT} = \text{incremental horizontal tail deflection (deg)}$$

$$\dot{h}_{msl} = \text{altitude above mean sea level (ft)}$$

For small angles, $\dot{h}_{msl}$ can be equated to aircraft vertical velocity which is $U_o(\theta - \alpha)$. The commanded input was pitch rate instead of normal load factor, and the required outputs of the system for feedback purposes were angle of attack, pitch rate, and normal load factor in units of g. The expression for normal load factor came from the Z-axis acceleration equation,

$$\begin{aligned} a_Z &= a_{Z_{cg}} - X_a\dot{q} \\ &= \dot{w} - q\,U_O - X_a\dot{q} \end{aligned} \tag{2.3}$$

where

$$a_Z = \text{Z body axis acceleration (ft/s}^2\text{)}$$

$$\dot{w} = \text{body axis linear vertical acceleration (ft/s}^2\text{)}$$

$$X_a = \text{distance from cg to accelerometer (ft)}$$

$$\dot{q} = \text{pitch acceleration (rad/sec}^2\text{)}$$

$$U_O = \text{steady-state velocity along the X body axis (ft/s)}$$

Using small angle approximations

$$\dot{\alpha} = \dot{w}/U_O \tag{2.4}$$

hence,

$$a_Z = U_o\,(\dot{\alpha} - q) - X_a\dot{q}$$

The direction of the normal load factor vector is opposite that of the Z-acceleration term (3:446). Therefore, normal acceleration at the accelerometer location, in units of incremental g is

$$A_n = [-U_o(\dot{\alpha} - q) + X_a\dot{q}](1/32.2) \qquad (2.6)$$

where

A_n = incremental normal load factor (g)

$\dot{\alpha}$ = angle of attack rate (rad/s^2)

X_a = distance from cg to accelerometer (ft)

$\dot{q}$ = pitch acceleration (rad/sec^2)

q = pitch rate (rad/sec)

U_o = steady-state velocity (ft/s)

The value of X_a was 13.93 feet, which corresponds to the location of the accelerometer under the pilot's seat. The eigenvalues, or poles, of the completed open-loop longitudinal system and representative modes are listed in Table 2.2. Figure 2.3 shows the completed open-loop longitudinal state-space matrix. Note that the F-16 has a characteristic unstable short period which is stabilized using pitch rate feedback, while angle of attack and normal acceleration feedback are used to give a better response.

Table 2.2: Eigenvalues and Representative Modes of the F-16 Longitudinal Axis

Eigenvalue	Mode
$-.008627 \pm i\,0.0719$	Phugoid
1.90	Short Period
-4.35	Short Period
-20.0	Actuator

$$\begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{\theta} \\ \dot{q} \\ \dot{\delta}_{HT} \\ \dot{h}_{msl} \end{bmatrix} = \begin{bmatrix} -.01485 & .6524 & -.5618 & -.3132 & .12255 & 0 \\ -.004786 & -1.4921 & -.0013 & .99278 & -.18817 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -.02063 & 9.7532 & .00029 & -.9591 & -19.041 & 0 \\ 0 & 0 & 0 & 0 & -20 & 0 \\ 0 & -11.6928 & 11.6928 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ \theta \\ q \\ \delta_{HT} \\ h_{msl} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 20 \\ 0 \end{bmatrix} [\delta_{HT}]_{cmd}$$

$$\begin{bmatrix} q \\ A_n \\ \alpha \\ h_{msl} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ .00158 & .61546 & .000475 & -.00462 & -.07541 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ \theta \\ q \\ \delta_{HT} \\ h_{msl} \end{bmatrix}$$

Figure 2.3: Open Loop Longitudinal State-Space System

Thus far, the state-space system is unstable and uses commanded horizontal tail deflection as the control input. However, by closing the feedforward and feedback paths shown in Figure 2.1, the system will become stable, and the commanded input will become pitch rate. The feedback and feedforward paths shown in Figures 2.1 and 2.2 can be expressed as a matrix in the Laplace domain in terms of aircraft outputs and inputs as shown in Figure 2.4.

$$\begin{bmatrix} \delta_{HT} \\ \delta_{Feff} \\ \delta_{R} \end{bmatrix}_{cmd} = \begin{bmatrix} 0 & \frac{1.076(s+4)(s+5)}{(s+1)(s+12)} & 0 & \frac{3.222(s+4)(s+5)}{s(s+12)} & 0 & \frac{5}{s+10} \\ \frac{6.0}{s+50} & 0 & 0 & 0 & 0 & 0 \\ \frac{.22116}{(s+50)} - \frac{3.0375\, s(s+5)}{(s+1)(s+15)(s+50)} & 0 & \frac{112.5\, s(s+5)}{(s+1)(s+15)(s+50)} & 0 & 9.66 & 0 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \\ A_n \\ A_y \\ \alpha \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & \frac{-23.4\,(s+5)}{s(s+60)} \\ \frac{-6.0}{s+50} & 0 \\ \frac{-.22116}{s+50} & 0 \end{bmatrix} \begin{bmatrix} p_{cmd} \\ q_{cmd} \end{bmatrix}$$

Figure 2.4: Feedback Matrix in the Laplace Domain

$$\delta_{HTcmd} = \left[\frac{1.076(s+4)(s+5)}{(s+1)(s+12)} \quad \frac{3.222(s+4)(s+5)}{s(s+12)} \quad \frac{5}{s+10} \right] \begin{bmatrix} q \\ A_n \\ \alpha \end{bmatrix} + \left[\frac{-23.4\,(s+5)}{s(s+60)} \right] q_{cmd} \tag{2.7}$$

Closed-Loop System Derivation. In order to build the closed-loop system, the feedback and feedforward paths must be transformed from the Laplace domain to the time domain. This was accomplished by putting each Laplacian element into a state-space phase-variable canonical form (5:210-215). Each of these individual matrices were then combined to form a state-space representation of the feedback and feedforward paths. Although this does not represent a minimal realization of the Laplacian matrix, it is, however, more intuitive and easily understood. The longitudinal feedback and feedforward state-space representations are shown in Appendix E.

In developing the closed-loop system, several unconventional aspects in the F-16 control system were encountered; most notable being that the F-16 utilizes negative input and positive feedback in its control law diagram. The aircraft open-loop transfer functions, which can be generated from the open loop system, have an overall negative sign associated with them due to the sign convention defining a positive horizontal tail deflection as being trailing edge down. If this negative sign is taken into account, then the control system will have the more traditional sign convention of negative feedback. When generating a state-space system using a computer program, negative feedback is usually assumed which means the state-space system must be properly set up if positive feedback is desired. This is the rationale for the negative signs that appear in the 'C' matrix of the feedback system.

Once the aircraft longitudinal plant, feedback, and feedforward matrices were developed, they were combined to form the closed-loop system. The derivation of the closed loop longitudinal system was necessary to ensure that the computer program was

building the proper system. Two controls analysis computer programs were utilized in this thesis: Comprehensive Control (CC) and MATRIX$_X$ (see References 6 and 7). Because it was able to work with both Laplace and state-space representations, CC was used initially to develop the aircraft transfer functions and transform the feedback and feedforward matrices into state-space form. Although it was a more intuitive program, CC was limited in the size of systems that it could handle and was very time consuming when determining output responses. Therefore, MATRIX$_X$ was used to form the combined longitudinal and lateral-directional closed-loop system, and also to evaluate the results of the optimization process.

Figure 2.5 shows a representation of the closed-loop control system with blocks E and K representing the feedforward and feedback matrices respectively. The state-space format for the open loop aircraft is represented by the following equations:

$$\dot{\underline{x}} = A\underline{x} + B\underline{u} \tag{2.8a}$$

$$\underline{y} = C\underline{x} \tag{2.8b}$$

The feedback system can be written as

$$\dot{\underline{x}}_k = A_k \underline{x}_k + B_k \underline{y} \tag{2.9a}$$

$$\underline{y}_k = \underline{u}' = C_k \underline{x}_k + D_k \underline{y} \tag{2.9b}$$

and the feedforward system as

$$\dot{\underline{x}}_E = A_E \underline{x}_E + B_E \underline{\delta}_{cmd} \tag{2.10a}$$

$$\underline{y}_E = \underline{u}'' = C_E \underline{x}_E + D_E \underline{\delta}_{cmd} \tag{2.10b}$$

where

$$\underline{\delta}_{cmd} = [\, q_{cmd} \quad p_{cmd} \,]^T$$

The plant input, u, is expressed as

$$\underline{u} = \underline{u}' + \underline{u}'' \tag{2.11}$$

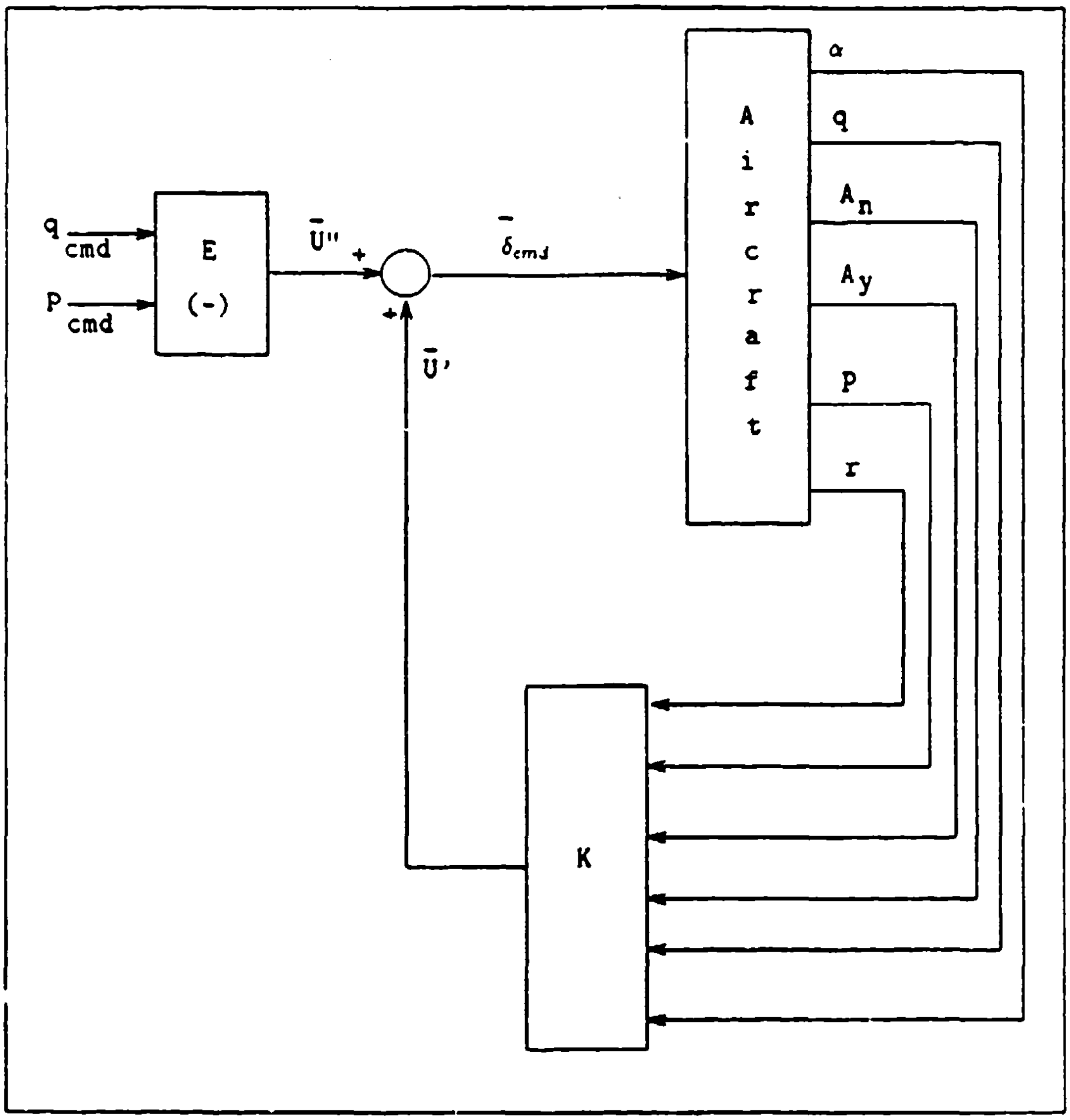


Figure 2.5: General Closed Loop State-Space System

Substituting Eq (2.8b) into Eqs (2.9a) and (2.9b) yields

$$\dot{\underline{x}}_k = A_k \underline{x}_k + B_k C\underline{x} \qquad (2.12a)$$

$$u' = C_k \underline{x} + D_k C\underline{x} \qquad (2.12b)$$

Placing (2.11) into (2.8a) results in the following equation:

$$\dot{\underline{x}} = A\underline{x} + B\underline{u}' + B\underline{u}'' \qquad (2.13)$$

Substituting (2.10b) and (2.12b) into (2.13) yields the expression:

$$\begin{aligned}\dot{\underline{x}} &= A\underline{x} + BC_k \underline{x}_k + BD_k C \underline{x} + BC_E \underline{x}_E + BD_E \delta_{cmd} \\ &= (A + BD_k C)\underline{x} + BC_k \underline{x}_k + BC_E \underline{x}_E + BD_E \delta_{cmd} \qquad (2.14)\end{aligned}$$

Collecting expressions for each of the state-space subsystems results in the following equations:

$$\dot{\underline{x}}_E = A_E \underline{x}_E + B_E \delta_{cmd} \qquad (2.10)$$

$$\dot{\underline{x}} = (A + BD_k C)\underline{x} + BC_k \underline{x}_k + BC_E \underline{x}_E + BD_E \delta_{cmd} \qquad (2.14)$$

$$\dot{\underline{x}}_k = A_k \underline{x}_k + B_k C\underline{x} \qquad (2.12a)$$

$$\underline{y} = C\underline{x} \qquad (2.8b)$$

These equations may now be combined to form a closed loop system represented by the following matrix:

$$\begin{bmatrix} \dot{\underline{x}}_E \\ \dot{\underline{x}} \\ \dot{\underline{x}}_k \end{bmatrix} = \begin{bmatrix} A_E & 0 & 0 \\ BC_E & A + BD_k C & BC_k \\ 0 & B_k C & A_k \end{bmatrix} \begin{bmatrix} \underline{x}_E \\ \underline{x} \\ \underline{x}_k \end{bmatrix} + \begin{bmatrix} B_E \\ BD_E \\ 0 \end{bmatrix} [\delta]_{cmd} \qquad (2.15)$$

$$\underline{y} = \begin{bmatrix} 0 & C & 0 \end{bmatrix} \begin{bmatrix} \underline{x}_E \\ \underline{x} \\ \underline{x}_k \end{bmatrix} \qquad (2.16)$$

The combined longitudinal and lateral-directional plant, feedback, feedforward, and closed loop state-space systems for the F-16 are shown in Appendix E. The above derivation is valid for any generic system and is not specifically intended for the control system presented in this study.

Table 2.3 presents the closed loop poles of the longitudinal system. Note that all of the poles are now stable with the short period mode having a damping coefficient of 0.723. The roots of the phugoid mode still lie on the real axis for this flight condition, and, therefore, do not cause any of the normal oscillatory motions of the phugoid mode.

Table 2.3: F-16 Longitudinal Closed Loop Poles

Eigenvalue	Mode
0 (x3)	h_{msl}, $\dot{h}_{msl}$, $\dot{\theta}$
-.01485	
-.64155	Phugoid
-2.1112	Phugoid
-3.3356 ± i 3.1843	Short Period
-10.2819	
-12.0	
-15.3023 ± i 15.6413	Actuators
-60.0	Pitch Rate Filter

The time responses of pitch rate, normal load factor, angle of attack, and aircraft altitude to a step pitch rate input are displayed in Figure 2.6. Note that the commanded input of the original control law was normal load factor and that the input of the autopilot has been changed to pitch rate using Eq (2.2). This change merely acts as a gain which changes the magnitude but not the shape of the aircraft time response.

Lateral-Directional Axis. The states used to build the lateral-directional state-space system were sideslip angle, heading angle, bank angle, roll rate, yaw rate, flaperon deflection, and rudder deflection:

$$\underline{X} = [\ \beta\ \ \psi\ \ \phi\ \ p\ \ r\ \ \delta_F\ \ \delta_R\ ]^T$$

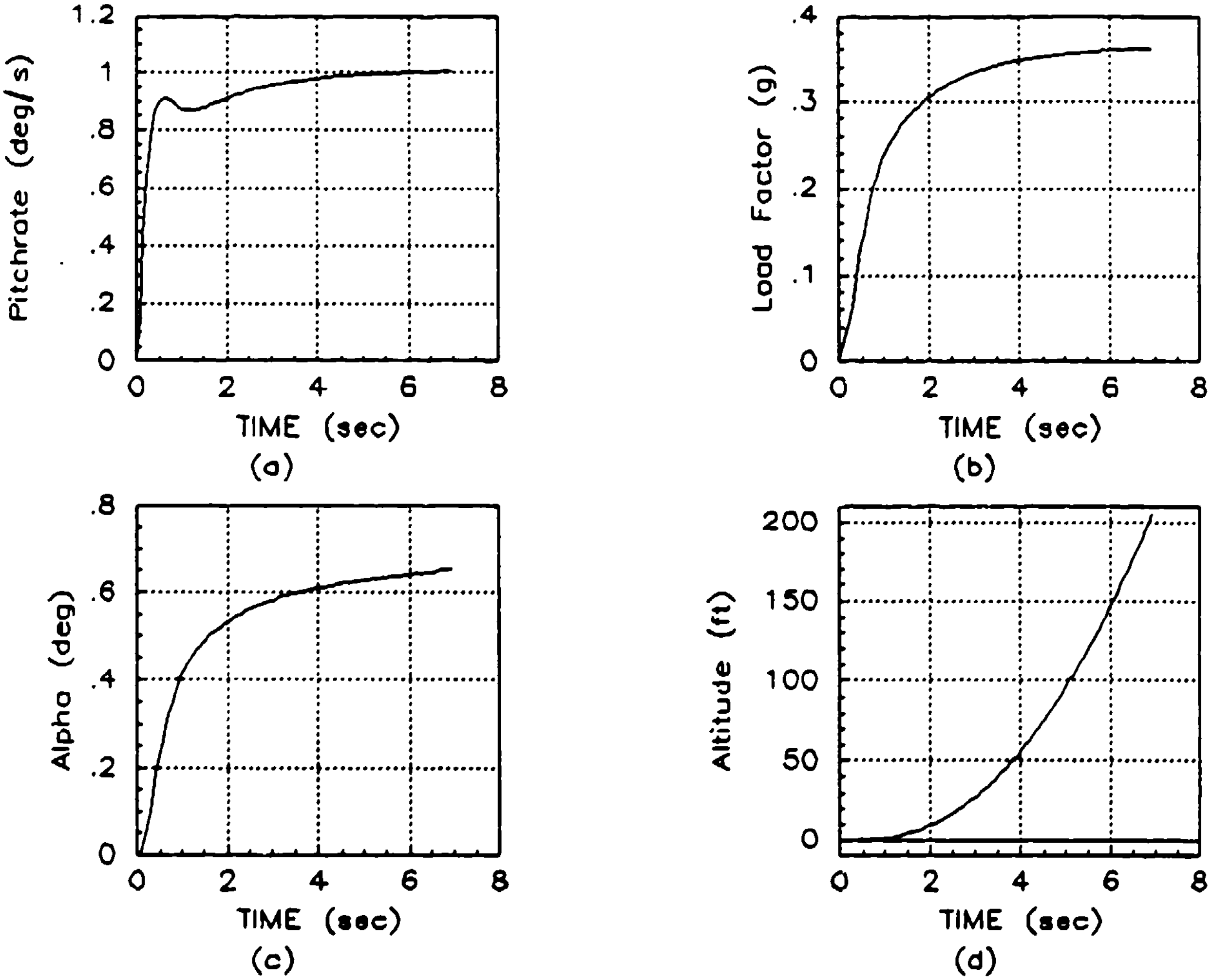


Figure 2.6: Aircraft Longitudinal State Responses To Step Pitch Rate Command Input: (a) Pitch Rate, (b) Normal Load Factor, (c) Angle of Attack, (d) Altitude

where

β = sideslip angle (deg)

ψ = heading angle (deg)

ϕ = bank angle (deg)

p = roll rate (deg/s)

r = yaw rate (deg/s)

δ_F = flaperon deflection (deg)

δ_R = rudder deflection (deg)

Roll rate was used as the input to the system, and the required outputs for system feedback were roll rate, yaw rate, and lateral load factor. Other outputs were eventually added to examine the aircraft response to various roll rate inputs. The derivation for lateral load factor came from the y-axis acceleration equation:

$$a_Y = a_{Y_{cg}} + X_a \dot{r}$$
$$= \dot{v} + r\,U_o + X_a\dot{r} \tag{2.17}$$

Again, from small angle approximations

$$\dot{\beta} = \dot{v} / U_o \tag{2.18}$$

then

$$a_Y = [\,U_o\,(\,\dot{\beta} + r) \;+\; \dot{r}\,X_a\,]\,[\,1/32.2\,] \tag{2.19}$$

where

a_y = lateral acceleration (g)

$\dot{\beta}$ = sideslip rate (rad/s)

r = yaw rate (rad/sec)

$\dot{r}$ = yaw acceleration (rad/s^2)

X_a = accelerometer distance from c.g. (ft)

Appendix E contains the open loop lateral-directional state-space matrix. The poles of the system and representative modes are listed below in Table 2.4.

Table 2.4: Eigenvalues of Lateral-Directional Axis

Eigenvalue	Mode
-0.08223	spiral
-2.45040	roll
$-.60237 \pm i\,2.92685$	dutch roll

The roots for these modes were confirmed using the equations for the roll and dutch roll approximations and were found to be in close agreement (3:367-377). This resulted in a roll mode time constant of 0.408 seconds, and a dutch roll natural frequency and damping coefficient of 2.9882 rad/s and 0.202. Thus, for the stated initial conditions, the lateral-directional axis of the F-16 model is stable but has the characteristic light dutch roll damping of most aircraft.

The analysis of the feedback paths for the lateral-directional system was performed in the same manner as that of the longitudinal axis. A phase-variable canonical state-space representation of the Laplace domain feedback and feedforward matrices is shown in Appendix E. The lateral-directional control system, previously seen in Figure 2.2 utilized $\dot{\beta}$ feedback for the yaw damper design which can be confirmed using the lateral acceleration equation:

$$a_Y = \dot{v} + ur - wp \tag{2.20}$$

and the substitutions

$$\dot{\beta} = \dot{v} / U_o \tag{2.18}$$

$$\alpha = w / U_o \tag{2.4}$$

The aileron-rudder interconnect (ARI) was linearized about the initial conditions, and a value of 0.03686 was selected for the trim angle of attack. Since the value of the ARI is dependent upon angle-of-attack, a mid-range value of AOA could have been selected if rolling maneuvers were going to be performed that represented a compromise between the 1-g initial condition and the 5-g maximum allowable load factor.

Construction of the closed loop lateral-directional control system followed the derivation used in the previous section. The closed loop poles were stable and well damped, and a time history of the aircraft response to a step roll rate input, seen in Figure 2.7, shows that the yaw damper worked properly by attempting to null out yaw rate and lateral acceleration. Figure 2.8 shows the control surface deflections for a step roll rate input. The roll rate response tapers off after reaching a peak value and does not have the characteristic exponential rise to a steady-state value for a reasonable time period as might be expected. The cause for this response is linked to the value of the closed loop spiral mode which is equal to -0.0123. This value can be traced to the magnitude of the open loop spiral mode root which has a value of -.0820. An examination of some open loop spiral mode roots for other aircraft revealed that this was a very large value. The Douglas A-4D has a spiral mode root of -.0060 at M = 0.6 and 15000 feet; 14 times smaller than that of the F-16 at sea level and the same Mach number (3: 700-706). The F-16 transfer function for roll rate to flaperon deflection shows that the spiral root is the primary cause of the uncharacteristic aircraft roll rate response:

$$\frac{P}{\delta_{F_{cmd}}} = \frac{-1291.93\, s\, [\, s + (.63593 \pm i\, 2.99211)]}{(s+20)\,(s+2.4504)\,(s+.08223)\,[\, s + (.60237 \pm i\, 2.92685)]} \tag{2.21}$$

Note that the complex conjugate zero nearly cancels out the dutch roll mode so that only the spiral and roll modes along with an actuator root are left in the denominator. Normally, the small value of the spiral mode will cancel the free s in the numerator for the time interval used to evaluate the roll rate response of the aircraft. This leaves only

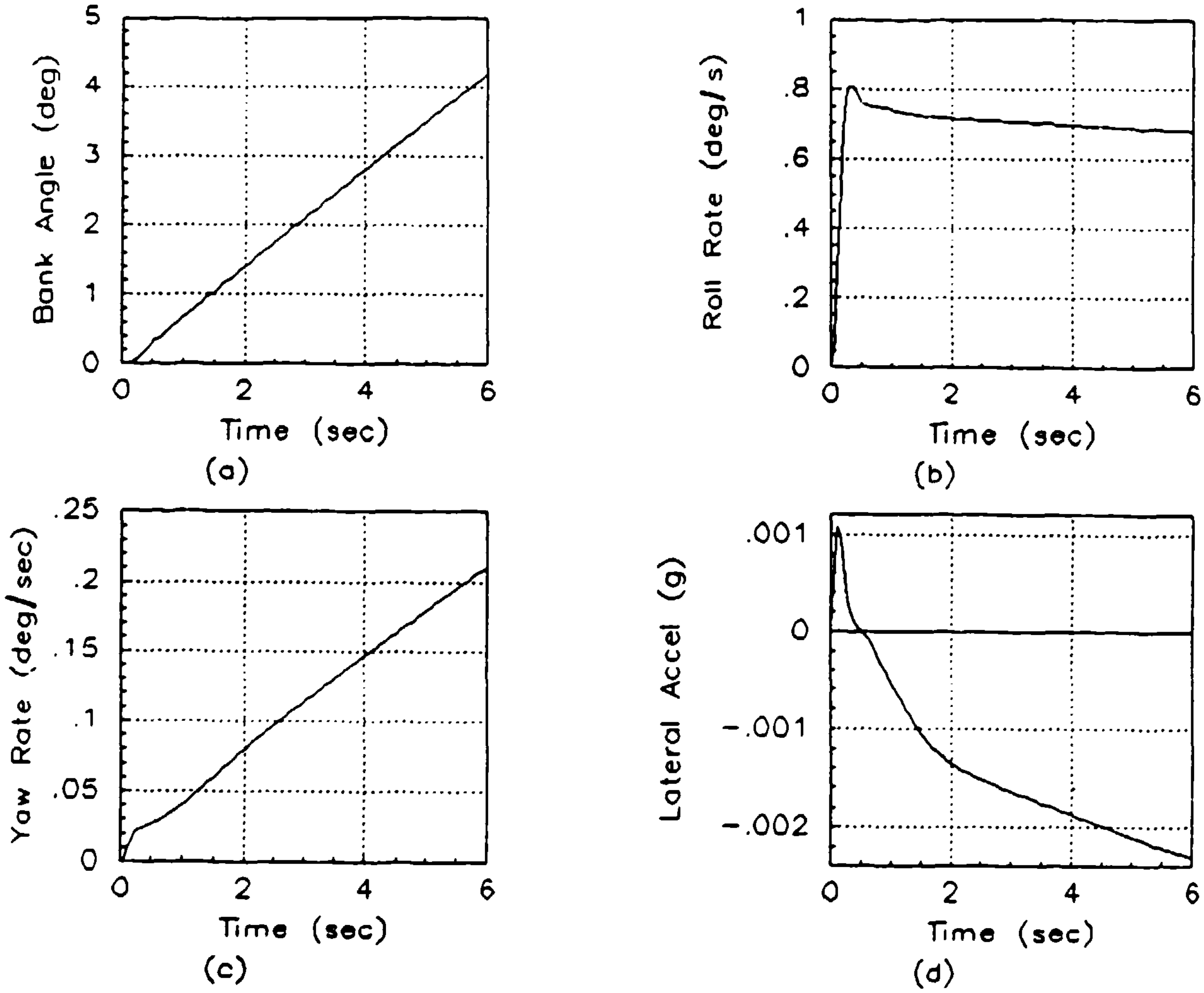


Figure 2.7: Aircraft Lateral-Directional State Responses To Step Roll Rate Command Input: (a) Bank Angle, (b) Roll Rate, (c) Yaw Rate, (d) Lateral Acceleration

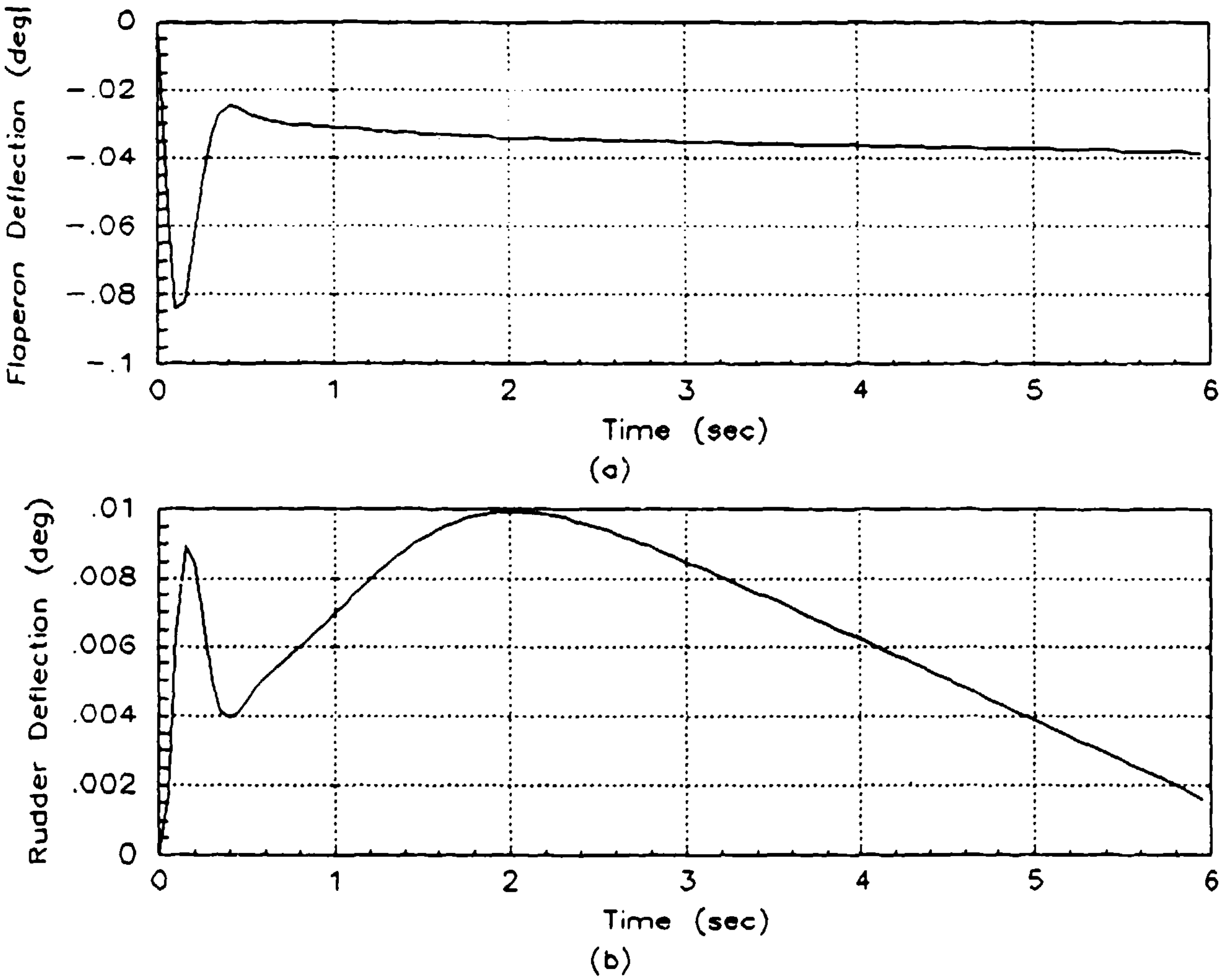


Figure 2.8: Lateral-Directional Control Surface Response to a Step Roll Rate Input: (a) Flaperon, (b) Rudder

the roll mode in the denominator which results in the characteristic exponential rise to a steady-state value for the roll rate response. The large magnitude of the F-16 spiral mode makes this assumption invalid and causes the response that is shown in Figure 2.7. To illustrate the pronounced effect the spiral root can have on roll rate response, Figure 2.9 displays four different time histories: the F-16 with its normal open loop spiral root; an F-16 with a spiral root that is one-tenth the normal magnitude, -.00822; the closed loop F-16; and the open loop A-4D. The roll rate to commanded flaperon deflection transfer function for the A-4D is more characteristic of traditional lateral transfer functions:

$$\frac{P}{\delta_{F_{cmd}}} = \frac{21.302\,s\,[\,s + (.40954 \pm i\,4.4136)\,]}{(s+1.5348)\,(s+.005963)\,[\,s + (.3830 \pm i\,4.3182)]} \tag{2.22}$$

No explanation can be given for the uncharacteristic roll rate response of the F-16 that resulted from the state-space system. Normally, the combination of the lateral-directional feedback loops and the ARI move the spiral root close enough to the imaginary axis so that the resultant roll rate response is exponential. Although the closed loop spiral root, -.0123, is about seven times smaller than that of the open loop, -.08223, it still causes a degradation in the roll rate response as seen in Figure 2.9. All approximations made for the roll, dutch roll, and spiral modes show the roots to be correct based on the control derivatives that were used (3:367-377). A check was made on the values of the control derivatives, but no errors were detected. The primary derivative that determines the value of the spiral mode is normally $C_{l\beta}$, but a comparison made with other aircraft shows its value to be comparable. One very plausible explanation is that the bank angle and roll rate attained are outside of the linearization limits used to construct the system, thereby violating the assumptions for small angle approximations.

Closed loop roll rate response exhibited the same degradation seen in the open loop. While this was not a critical problem, a more serious side effect of the overly stable

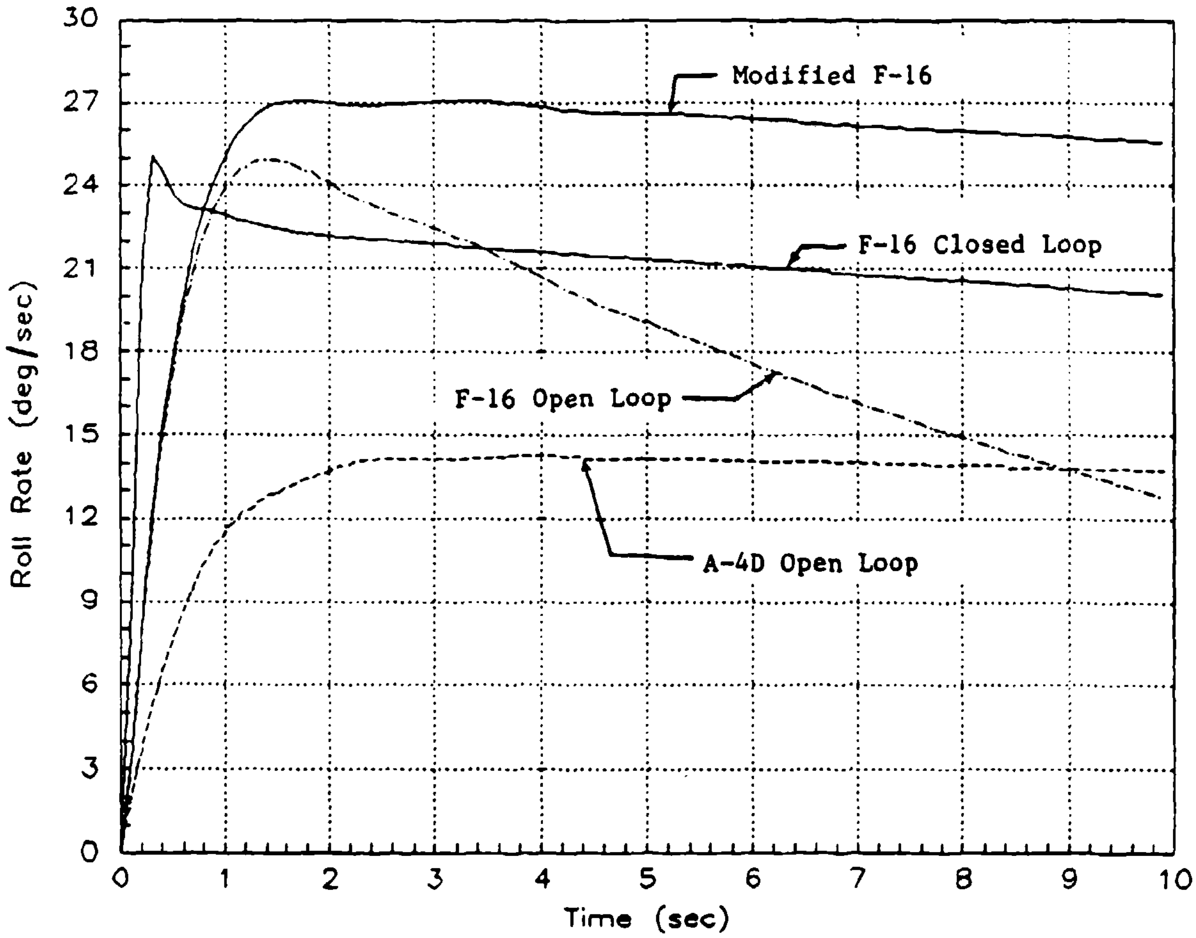


Figure 2.9: Roll Rate Response Comparisons Between the F-16 and A-4D

spiral mode was that the F-16 model could not be commanded to hold a constant bank angle. Figure 2.10 illustrates this problem by showing the response of the aircraft when initialized at 180 degrees of bank, ie., inverted. Figure 2.11 shows the flaperon and rudder time responses for this condition. Because of this phenomenon, any attempts to maneuver in the lateral-directional axis were ineffective. For example, when an aircraft is placed in a 60 degree bank and commands 2-g of normal load factor, the result will be a level turn. However, the model began rolling to a wings-level attitude which resulted in a climbing, 2-g turn. Because of these problems with the lateral-directional axis, the scope of the development for the ground collision avoidance system will be restricted to the longitudinal axis. This will be dealt with in more detail in Chapter 3.

State-Space Verification Using Sequential Loop Closure

Before proceeding any further in the development of the optimization process, a quick confirmation of the closed loop system should be performed using sequential loop closure and transfer functions to ensure that the state-space matrix is correct. Only the longitudinal axis will be verified in this case since it is the most critical component. The longitudinal control system, previously shown in Figure 2.1, can be redrawn to look like that pictured in Figure 2.12. Using the longitudinal open loop state-space matrix, the transfer functions for $\alpha(s)/\delta_{HT}(s)$, $q(s)/\delta_{HT}(s)$, and $A_n(s)/\delta_{HT}(s)$ can be derived from the equation

$$G(s) = C(sI-A)^{-1}B + D \tag{2.23}$$

where G(s) is the transfer function and A, B, C, and D are the matrices of the state-space system. The resulting open loop transfer functions are then represented by the following equations:

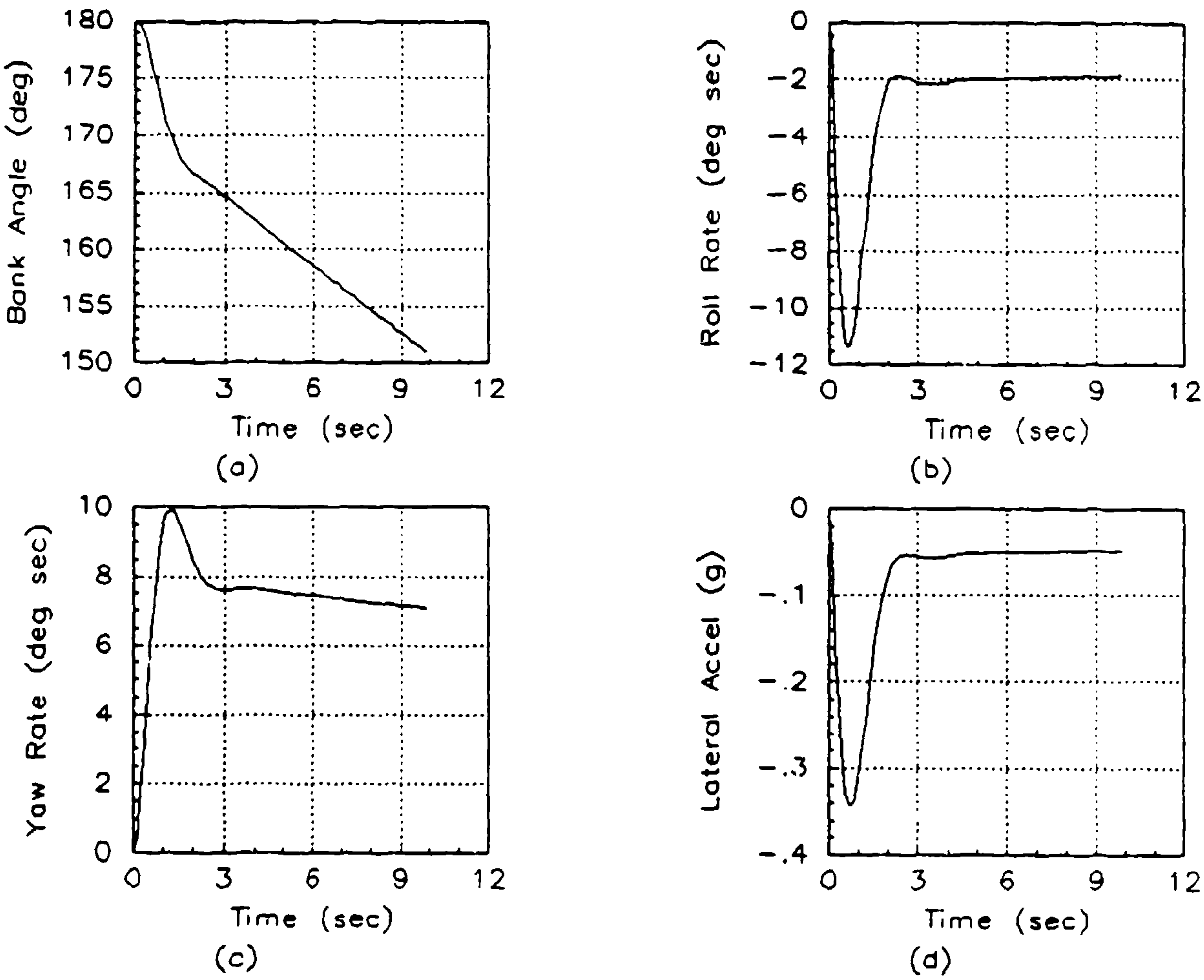


Figure 2.10: Aircraft Lateral-Directional State Responses Given Initial Bank Angle Condition of 180 degrees : (a) Bank Angle, (b) Roll Rate, (c) Yaw Rate, (d) Lateral Acceleration

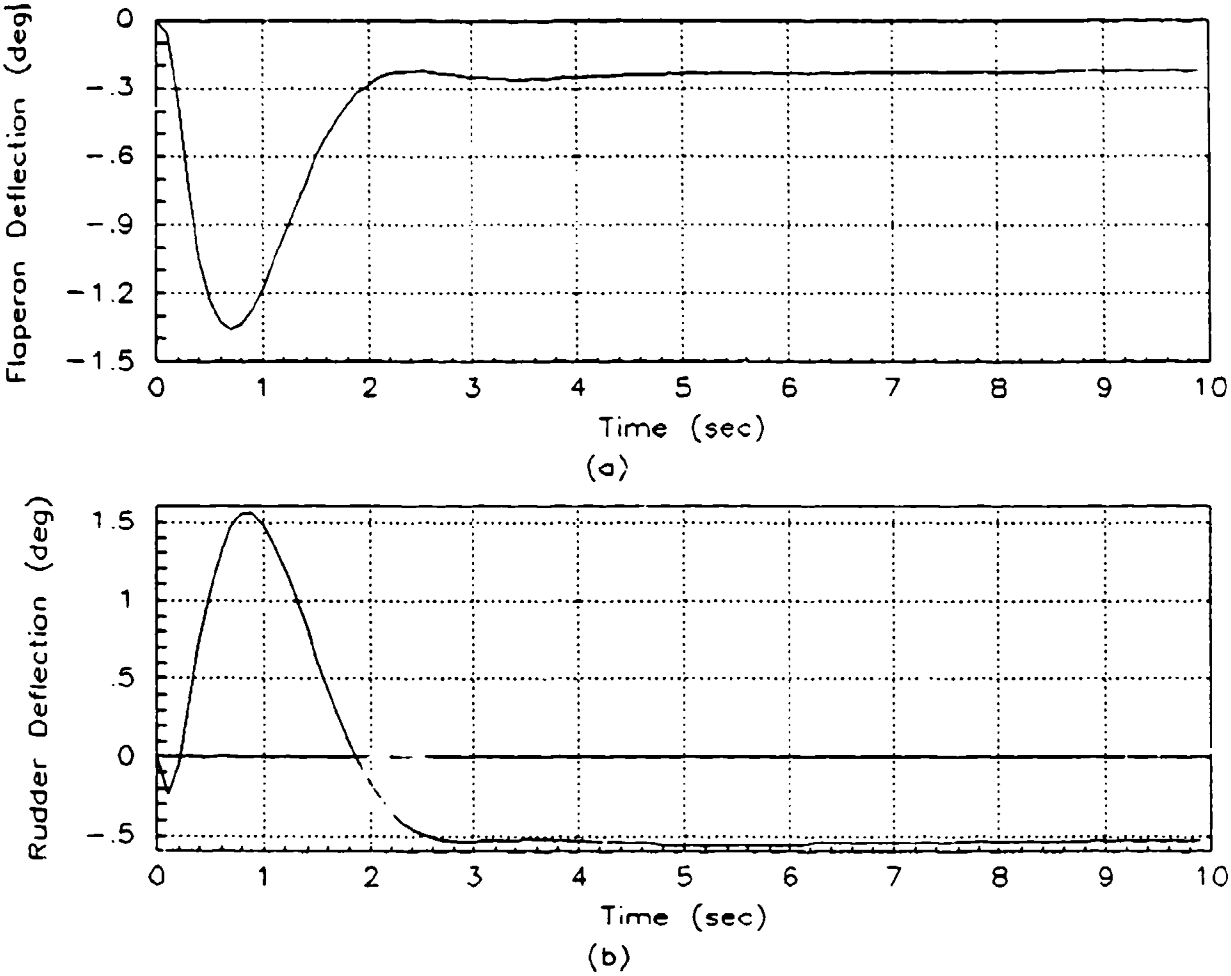


Figure 2.11: Control Surface Response for 180 degree Initial Conditions: (a) Flaperon, (b) Rudder

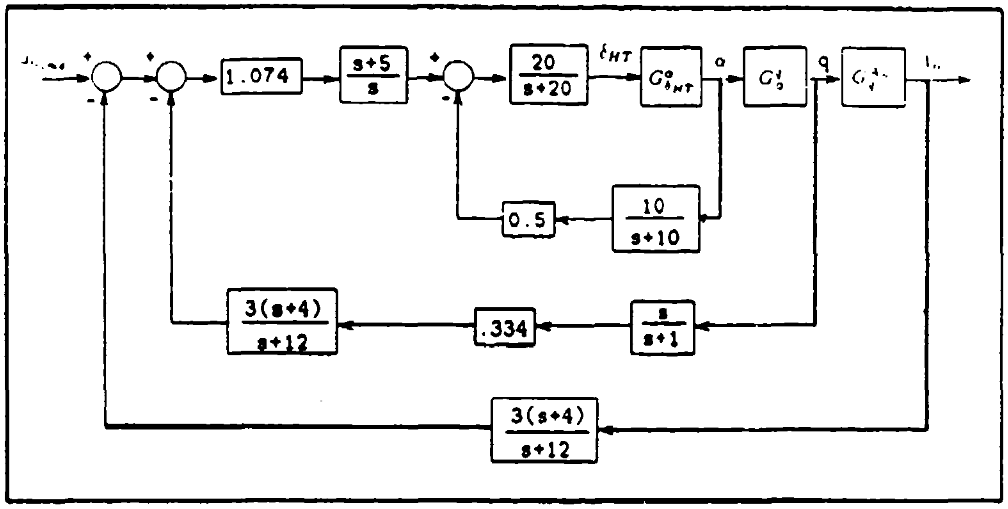


Figure 2.12: F-16 Longitudinal Control System Displayed in Loop Form

$$\frac{\alpha(s)}{\delta_{HT}(s)} = \frac{.18817\,(s + 101.422)\,[s + (.00756 \pm i\,0.04990)]}{(s + 4.349)\,(s - 1.901)\,[s + (.00864 \pm i\,0.0720)]} \tag{2.24}$$

$$\frac{q(s)}{\delta_{HT}(s)} = \frac{19.0412\,s\,(s + .01707)\,(s + 1.5864)}{(s + 4.349)\,(s - 1.901)\,[s + (.00864 \pm i\,0.0720)]} \tag{2.25}$$

$$\frac{A_n(s)}{\delta_{HT}(s)} = \frac{.07543\,s\,(s + .01544)\,[s + (1.40876 \pm i\,11.9844)]}{(s + 4.349)\,(s - 1.901)\,[s + (.00864 \pm i\,0.0720)]} \tag{2.26}$$

where $\alpha(s)$, $q(s)$, and $\delta_{HT}(s)$ are in degrees and $A_n(s)$ is in units of g.

Note that the negative sign associated with these transfer functions has been omitted, and instead used to provide negative feedback in the control loop (4: 1165-1177).

When performing sequential loop closures, the closing process starts with the inner loops and works towards the outer loops. Therefore, closing the angle of attack feedback loop first results in:

$$\frac{\alpha(s)}{\alpha(s)_{cmd}} = \frac{3.76342\,(s+10)\,(s+101.42)\,[s + (.00756 \pm i\,0.04990)]}{(s+19.5)\,(s+11.98)\,(s+.1326)\,(s-.1115)\,[s+(.4821 \pm i\,0.9371)]} \tag{2.27}$$

Note that the system is still unstable for the selected flight conditions. To further improve on the stability and increase the damping, pitch rate will be fed back in the next loop. In order to proceed to the next stage of loop closure, the forward path must be changed to the transfer function $q(s)/\alpha(s)_{cmd}$. This is accomplished by multiplying the ratio of the numerators of Eqs (2.25) and (2.24) by Eq (2.27):

$$\frac{q(s)}{\alpha(s)_{cmd}} = \frac{\alpha(s)}{\alpha(s)_{cmd}} \cdot \frac{q(s)}{\alpha(s)} \tag{2.28}$$

The pitch rate feedback loop is now closed, and the closed loop pitch rate transfer function is now formed:

$$\frac{q(s)}{q(s)cmd} = \frac{409.005\,(s+12)\,(s+1)\,(s+5)\,(s+10)\,(s+1.5864)}{(s-.0120)(s+.02885)(s+1.5202)\,[s + (3.3679 \pm i\,2.4190)]}$$

$$\cdot \frac{(s+.01707)}{[s + (13.4883 \pm i\,17.801)]\,(s+10.2226)} \tag{2.29}$$

The control ratio of $A_n(s)/q(s)_{cmd}$ is now formed by the same method used to form Eq (2.28):

$$\frac{A_n(s)}{q(s)_{cmd}} = \frac{q(s)}{q(s)_{cmd}} \cdot \frac{A_n(s)}{q(s)} \tag{2.30}$$

Closing the outer load factor loop will yield the transfer function for $A_n(s)/A_n(s)_{cmd}$, which is now stable and well damped:

$$\frac{A_n(s)}{A_n(s)_{cmd}} = \frac{1.6202\,(s+.01544)\,(s+1)\,(s+5)\,(s+10)\,(s+12)}{(s+.01509)\,(s+.6415)\,[s + (3.3358 \pm i\,3.1840)]}$$

$$\cdot \frac{[s + (1.4088 \pm i\,11.9844)]}{(s+2.1112)\,(s+10.2818)[s + (15.3028 \pm i\,15.6428)]} \tag{2.31}$$

Using Eq (2.31), any other response to the commanded input can be derived using ratios of equations similar to Eqs (2.28) and (2.30).

The time response of the closed loop transfer function, Eq (2.31), can now be compared to the response of the closed loop state-space system. A comparison showed that both responses were identical which indicates that the state-space system is correct. This was also verified using Reference 4. In addition, the poles of Eq (2.31) closely match the eigenvalues of the longitudinal state-space system. A similar but more complicated analysis can be performed for the lateral-directional axis if desired. However, since this axis is traditionally not as critical, the analysis will not be performed in this thesis.

III. Terrain Avoidance Control System Development

The purpose of this section will be to develop the theory and control system required to implement a terrain avoidance system. This design and development will be based on the capability of the digital terrain database to 'see' ahead of the aircraft and guide it over terrain obstacles. The theory for the altitude control system will first be developed followed by the general design of the control system. Design of the specific loops of the control system will next be accomplished using the root locus method. Finally, a terrain model will be introduced for evaluation of the control system.

Terrain Avoidance Equation Derivation

The capabilities of the digital terrain database will afford small, fighter-type aircraft the ability to perform terrain following flight without a large forward-looking radar. Because the terrain data is digitized, a discrete distance ahead of the aircraft can be chosen for viewing the approaching terrain. By selecting two points ahead of the aircraft in addition to a point directly below the aircraft, an arc in the form of a parabola can be formed as depicted in Figure 3.1. The furthest point, called $h_g(3)$, is located a distance, d, ahead of the aircraft while the second point, labeled $h_g(2)$, is positioned at a distance of d/2. A parabolic equation is selected because it corresponds to a constant acceleration path, hence a commanded pitch rate or load factor. The form of the equation will then be represented by

$$f(x) = C_1x^2 + C_2x + C_3 \tag{3-1}$$

with the boundary conditions of

$$\begin{aligned} f(0) &= h_g(1) \\ f(d/2) &= h_g(2) \\ f(d) &= h_g(3) \end{aligned} \tag{3-2}$$

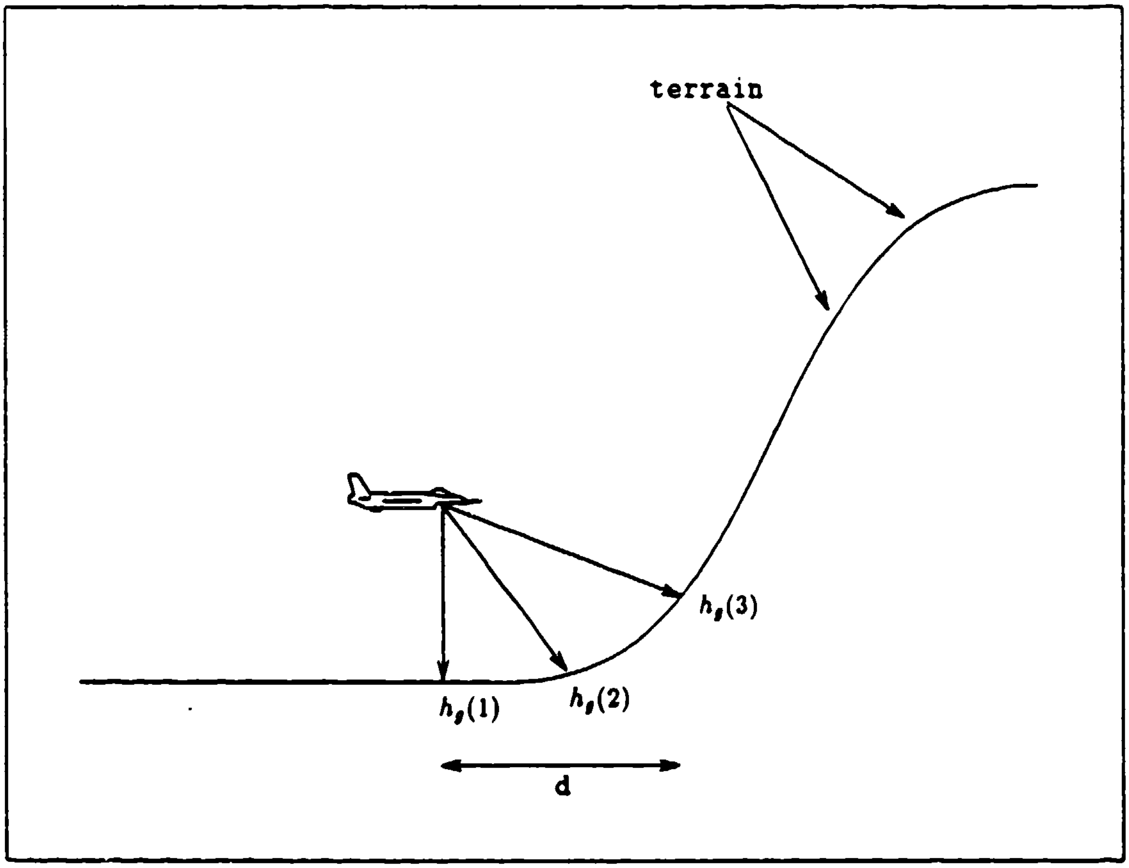


Figure 3.1: Scheme for Implementing Terrain Avoidance

Evaluating Eq (3-1) at the boundary conditions will result in

$$f(0) = C_3 = h_g(1) \tag{3-3}$$

$$f(d/2) = C_1 d^2/4 + C_2 d/2 + h_g(1) = h_g(2) \tag{3-4}$$

$$f(d) = C_1 d^2 + C_2 d + h_g(1) = h_g(3) \tag{3-5}$$

Solving Eqs (3-4) and (3-5) simultaneously will produce the value for C_1:

$$C_1 = 2\frac{[\,h_g(1) - 2h_g(2) + h_g(3)\,]}{d^2} \tag{3-6}$$

Substituting this value for C_1 back into Eq (3-5) will yield the value for the coefficient C_2:

$$C_2 = \frac{-3h_g(1) + 4h_g(2) - h_g(3)}{d} \tag{3-7}$$

To attach some physical meaning to the coefficients, aircraft states must be associated with the equations. The value of Eq (3-1) will yield an altitude, therefore aircraft altitude will become one of the states in the control system architecture. Evaluating Eq (3-1) at $x = 0$, which is directly below the aircraft will show that the value of the input for the altitude loop will be $h_g(1)$.

In order to avoid impacting the terrain, the velocity vector of the airplane must be aligned with the slope of the ground. By taking the derivative of Eq (3-1), the slope of the parabola will be given, and this value can then be set equal to the aircraft's flight-path angle. Taking the first derivative of Eq (3-1) and evaluating it at $x=0$ results in

$$f'(x)\big|_{x=0} = 2C_1(0) + C_2 = C_2 \tag{3-8}$$

Therefore, coefficient C_2 will be the input to the flight path angle loop of the altitude controller.

The second derivative of Eq (3-1) will give information concerning the curvature of the terrain. This curvature will be associated with the pitch rate of the aircraft which is also representative of the normal acceleration of the aircraft. Taking the second derivative of Eq (3-1) and evaluating it at x=0 will produce the required pitch rate input into the control system:

$$f''(x)\,|_{x=0} = 2C_1 \qquad (3\text{-}9)$$

A control law block diagram can now be drawn which will represent the general form of the control system before compensation is added. This diagram is shown on the following page in Figure 3.2. The 200 foot bias that is summed into the altitude loop is placed there for the purpose of keeping the aircraft 200 feet above the terrain during the avoidance maneuver. The F-16 will initially be at 200 feet, and should be at 200 feet at the end of the maneuver. By feeding back the output of the three aircraft states, an input error will be formed which will be the actual input into the aircraft plant. Note that the gains associated with each altitude input will be inversely proportional to the distance at which terrain is being viewed ahead of the aircraft. In the next section, the values for the compensators K_h, K_q, and K_q will be determined.

Control System Design Process

The design of the altitude control system will be performed using the root locus method for placing poles. In order to facilitate the understanding of the design process, Figure 3.2 has been redrawn to appear as a more conventional control system as shown in Figure 3.3. The design process will follow along the lines of sequential loop closure which was discussed in Chapter II. When designing the two inner-loop compensators, all external inputs to the system such as $h_s(2)$ will be set to zero. The outputs of each successive loop will be formed by using the ratio of the open-loop numerator of the

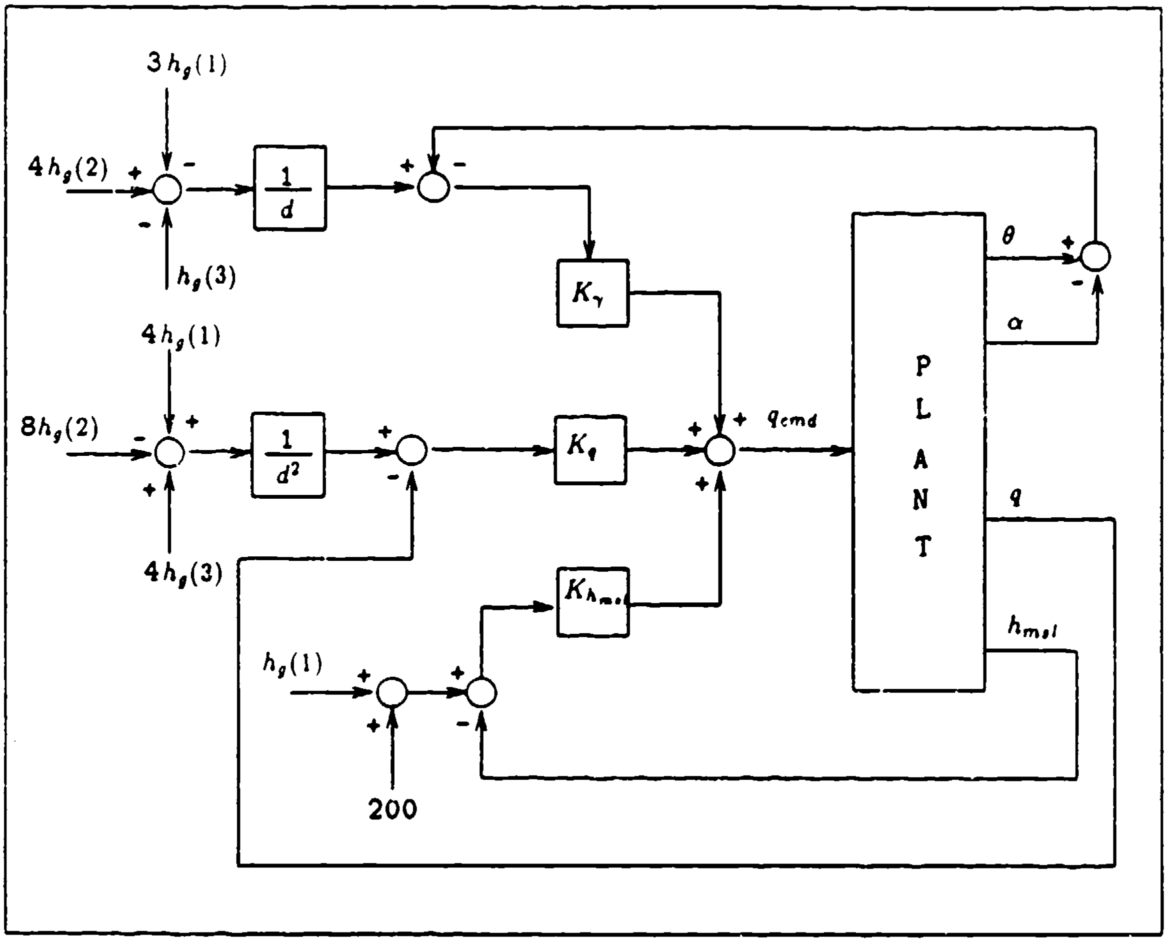


Figure 3.2: Terrain Avoidance Control System Diagram

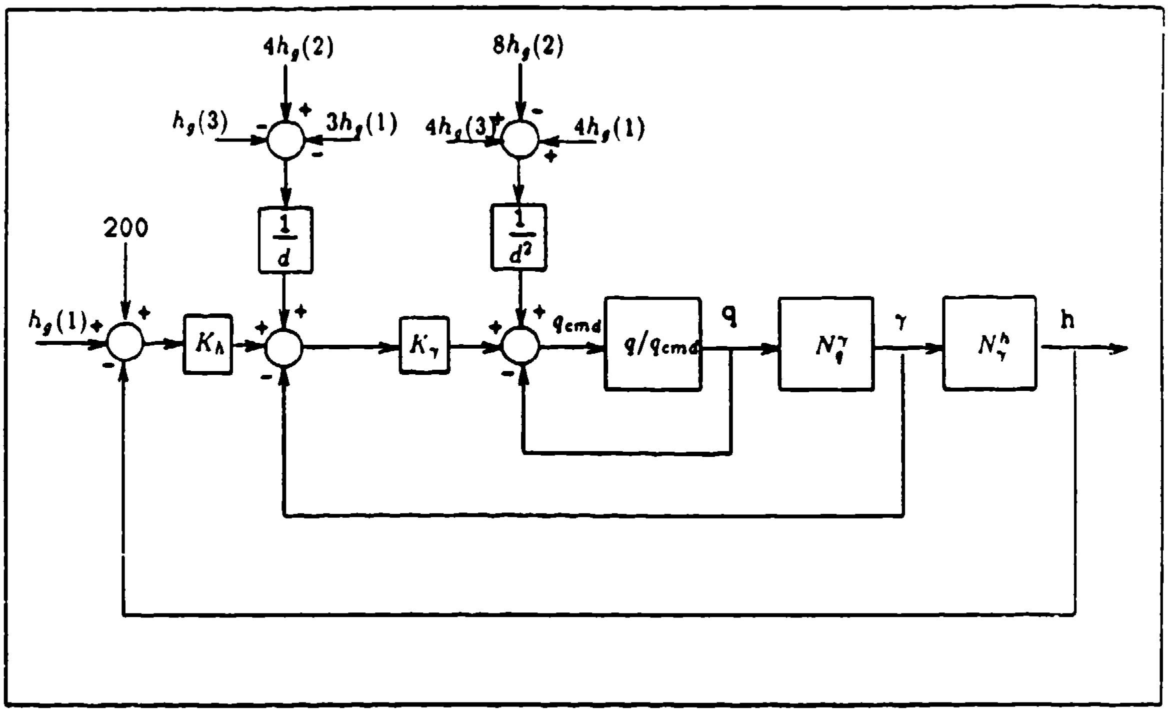


Figure 3.3: Terrain Avoidance Control System In Loop Form

desired output to that of the current output. To aid in the design process, only unity feedback will be used since the goal is to form an error signal.

Before stepping through the design process, the numerators of the output transfer functions will be re-introduced. They are as follows:

$$N^q = -380.82\, s^2 (s + 1.586)(s+.017068) \tag{3-10}$$

$$N^h = 44.005\, (s + .01544)(s + 12.794)(s - 12.563) \tag{3-11}$$

$$N^\gamma = 3.7634\, s\, (s + .01544)(s + 12.794)(s-12.563) \tag{3-12}$$

Note that both the altitude and flight path angle numerators contain a root in the right-half plane indicating that they are nonminimum phase in nature. This will affect the response of the aircraft to altitude inputs as will be shown in the next chapter.

The design process will begin by closing the inner-most loop of the controller, which is the pitch rate loop. The open loop pitch rate to pitch rate command transfer function of the controller is identical to the closed loop system that was derived for the aircraft in the previous section:

$$\frac{q(s)}{q(s)_{cmd}} = \frac{8911.2\, s\, (s + 1.5864)\, (s + .01707)\, (s + 1)\, (s + 5)}{(s + .01486)\, (s + .6416)\, (s + 2.1112)\, (s + 10.282)\, (s + 60)}$$

$$\cdot \frac{(s + 10)(s + 12)}{[\, s + (3.3356 \pm i\, 3.1843)][s + (15.3023 \pm i\, 15.6413)]} \qquad (3\text{-}13)$$

Since the pitch rate response of the aircraft is already satisfactory, no compensation is required. Therefore, pitch rate will just be fed back to form the pitch rate loop for the altitude controller:

$$\frac{q(s)}{q(s)_{cmd\ cl}} = \frac{8911.192\, s\, (s + 1.5864)\, (s + .01707)\, (s + 12)\, (s + 10)}{(s + .00011)(s + .01597)(s + .7691)(s + 1.9061)(s + 10.314)}$$

$$\cdot \frac{(s + 5)(s + 1)}{(s + 5.044 \pm i\, 3.1853)(s + 12.136 \pm i\, 18.815)(s + 62.958)} \qquad (3\text{-}15)$$

The pitch rate response of the aircraft to a step pitch rate command is shown in Figure 3.4.

Now that the pitch rate loop is closed, the flight path angle control loop can be designed. The open loop flight path to pitch rate command transfer function is formed by multiplying Eq (3-15) by the ratio of Eq (3-12) to Eq (3-10). Figure F.1 shows a plot of the root loci of this transfer function with no compensation added. The zero in the right-half plane is not shown due to scaling, however, its presence pulls one branch of the locus into the right-half plane. This has the effect of limiting the amount of gain

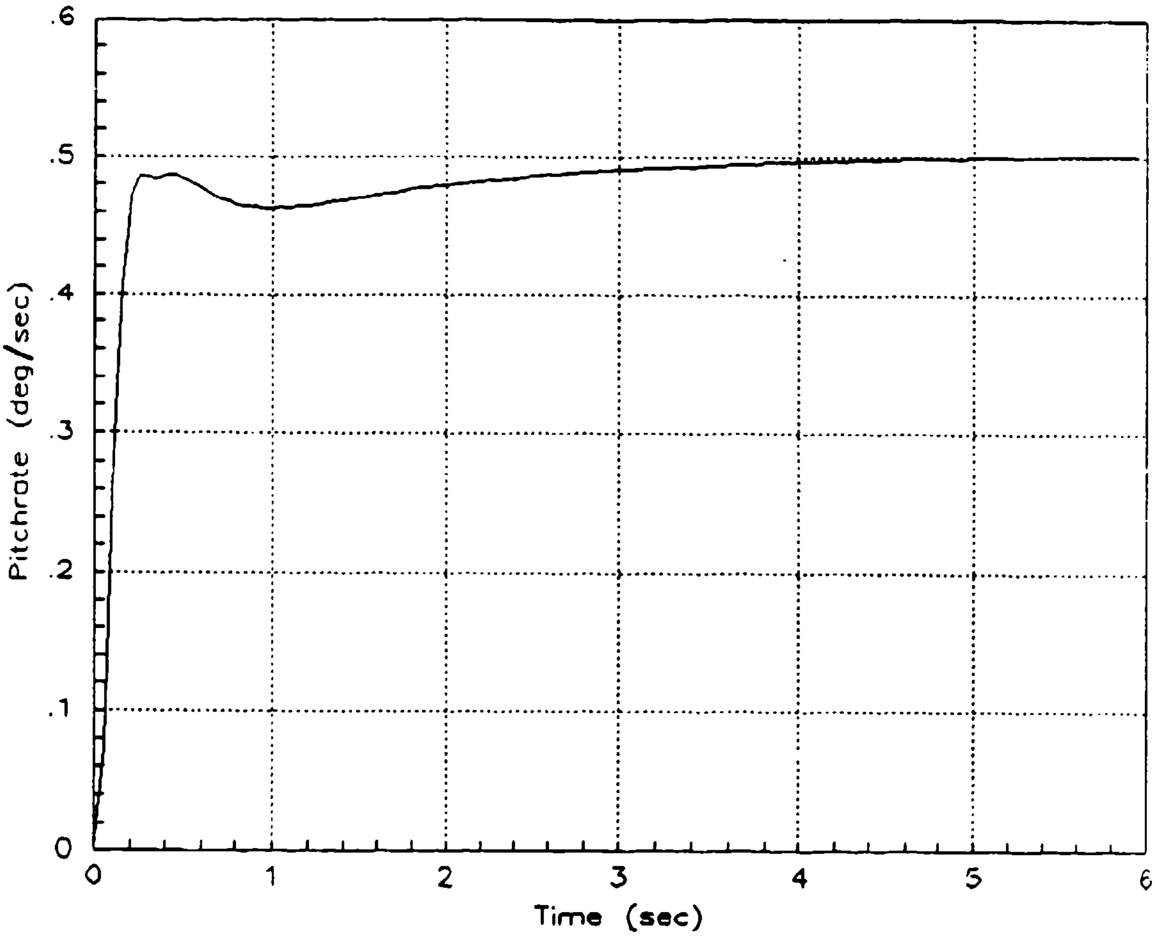


Figure 3.4: Aircraft Pitch Rate Response to Step Pitch Rate Command Input

that can be used to obtain a good response. For this reason, a lead compensator is required that will pull this branch over into the left-half plane.

The placement of the compensator zero was made such that the new branch formed on the real axis would attract the branch that originally split and crossed the imaginary axis. A zero value of -1.80 was selected, and the lead compensator became

$$K_\gamma = \frac{(s + 1.8)}{(s + 100)} \tag{3-16}$$

The effect of the compensator is shown in Figures F.2 and F.3. The poles furthest over in the left-half plane now migrate to the right-half plane zero, and the new branch formed by the placement of the zero on the real axis attracts the split branch closest to the imaginary axis. A gain is now selected that will locate the new poles further into the left-half plane. Figure F.3 displays the position of the new closed loop poles, indicated by the square boxes, for a gain of 200. Letting H represent the product of the gain times the compensator and G represent the plant, the closed loop transfer function will be represented by

$$Y(s) = \frac{GH}{1 + GH} \tag{3-17}$$

A confirmation on the effect of the lead compensator is shown by the time response plot in Figure 3.5. The aircraft flight path angle, γ, reaches 90 percent of its final value in approximately 1.4 seconds which is not outstanding, but does represent a good, stable response. The nonminimum phase nature of the system can also be seen in the first 0.20 seconds of the response.

Now that the closed loop flight path loop has been formed, the outer loop of the altitude controller can be designed. The open loop altitude to flight path angle

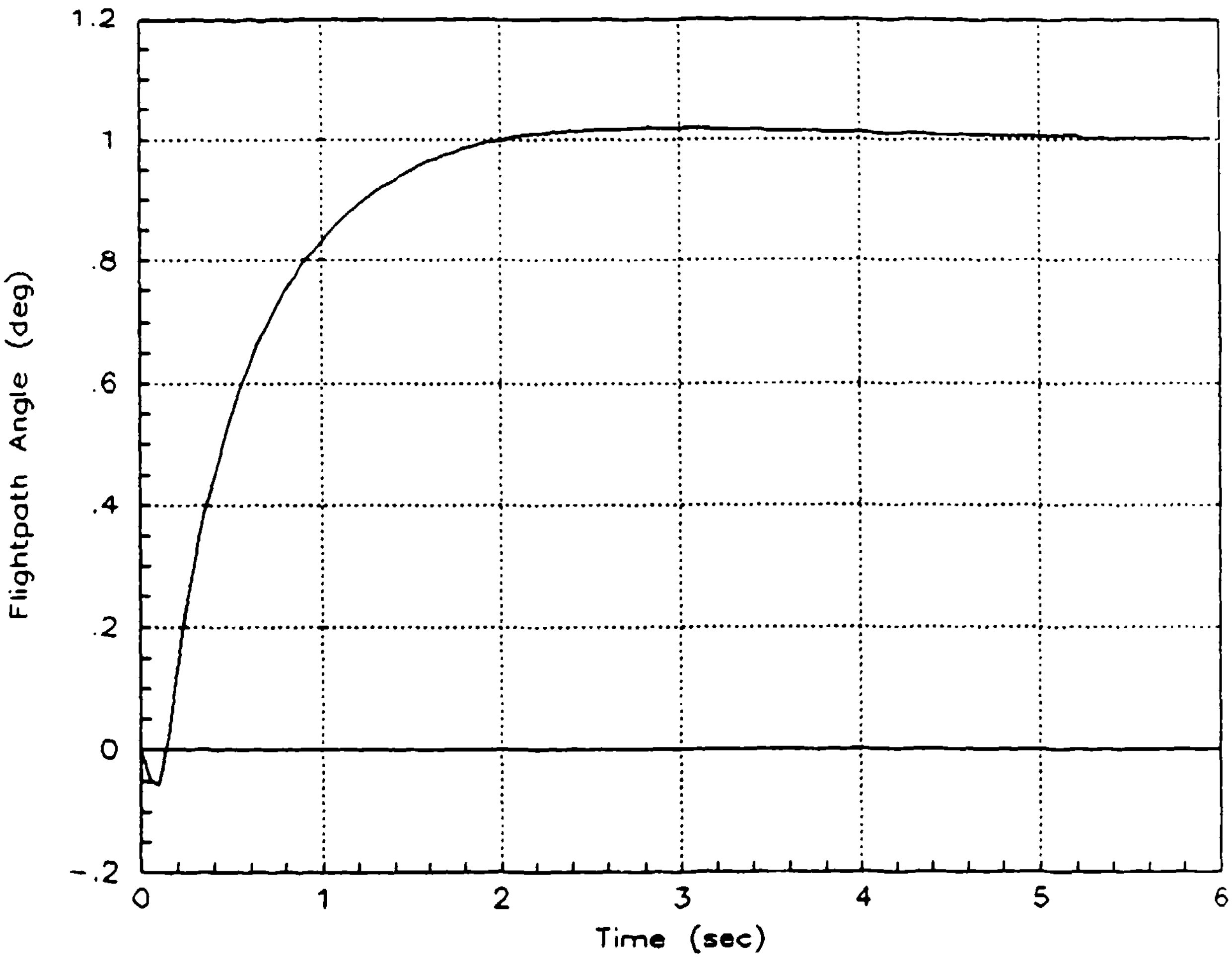


Figure 3.5: Aircraft Flight Path Angle Response to Step Flight Path Angle Command Input

command transfer function is formed using the ratio of numerators as previously discussed, and the root locus of this loop is shown in Figures F.4 and F.5. Note that the closed loop poles formed in the previous loop become the open loop poles of the current loop. Once again, due to the nonminimum phase of the altitude transfer function, the branch of the locus that is closest to the imaginary axis is migrating towards the right-half plane zero. Therefore, a lead compensator will also be required in this loop if a satisfactory response is to be achieved.

In order to move the poles that are closest to the imaginary axis further into the left-half plane, a zero will be placed to the left of the previous compensator zero. The compensator that will be used is

$$K_h = \frac{(s+2)}{(s+100)} \tag{3-18}$$

This will break the normal pole-zero branch and form a zero-zero branch, causing the complex-conjugate poles to migrate to the left instead of the right as depicted in Figure F.6. A larger view of the entire root locus is shown in Figure F.7. The gain selected for this loop was 15, and the location of the closed loop poles of the system are indicated by the boxes. Forming of the closed loop system is accomplished using Eq (3-17).

Using Figure 3.6 to evaluate system performance, the time history of the closed loop altitude controller shows that the system is well damped and exhibits an excellent rise time of approximately 0.45 seconds. The nonminimum phase portion of the response is also very evident in the first 0.2 seconds. This controller must now be put in a state-space format and integrated into the closed loop state-space system of the F-16 that has already been derived in Chapter II.

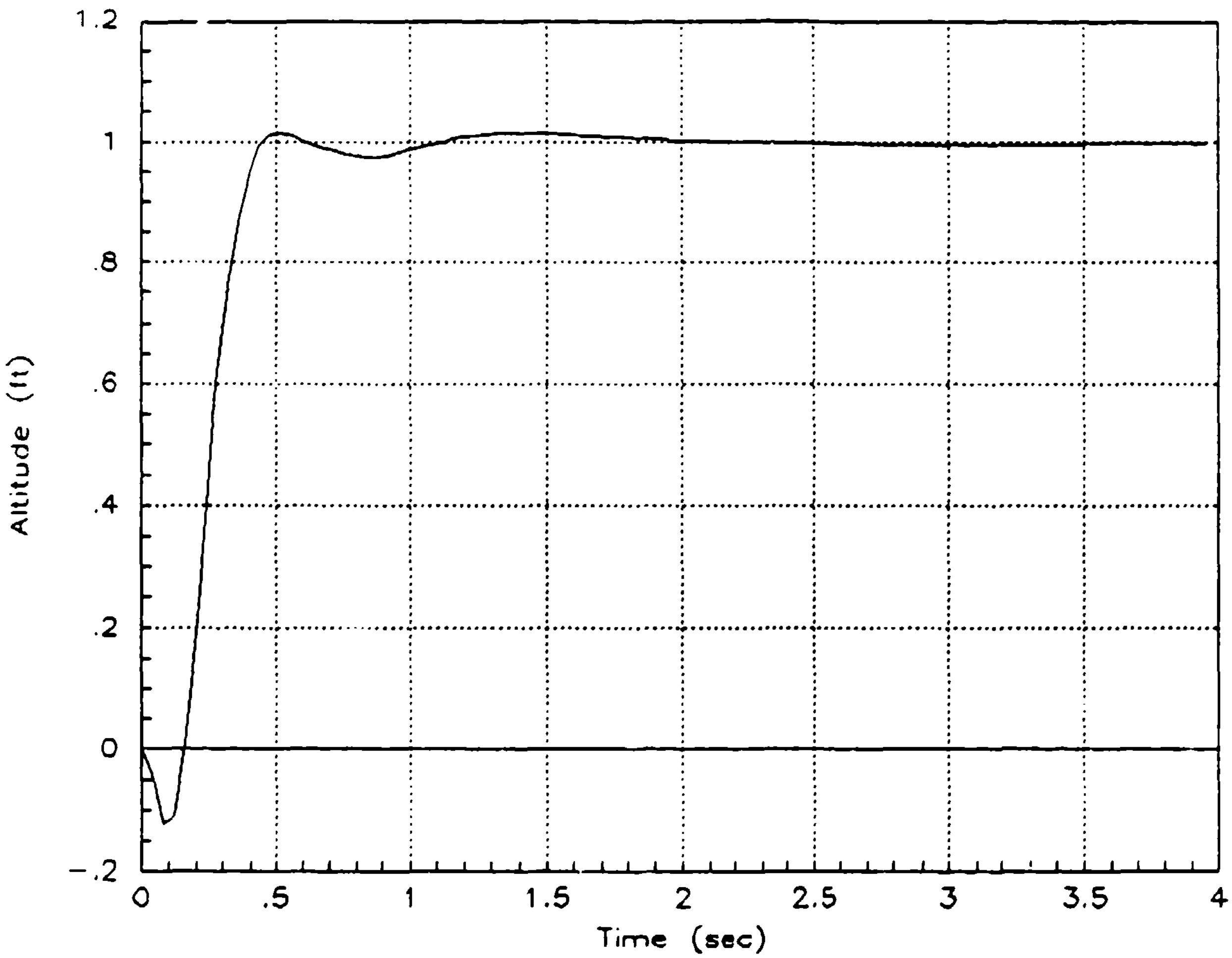


Figure 3.6: Aircraft Altitude Response to Step Altitude Command Input

Controller State-Space Derivation

Once the closed loop system has been derived it is necessary to express it in state-space format so that it can be combined with the state-space representation of the aircraft. The reason this must be done is that computer programs for control system analysis, such as MATRIX$_X$ which will be used here, require large systems to be placed in state-space format (Reference 7). The object of placing the controller in state-space form is to derive an expression for the pitch rate command to be input into the closed loop aircraft plant. This is seen more clearly by referring back to Figure 3.2.

If the inputs into the compensators, labeled K_γ, K_q, and K_h, are expressed as error signals and given the designations γ_{err}, q_{err}, and h_{err}, then an expression can be derived for pitch rate command:

$$q_{cmd} = [K_\gamma \;\; K_q \;\; K_h\,]\,[\gamma_{err} \;\; q_{err} \;\; h_{err}\,]^T \tag{3-18}$$

The error signals can then be expressed as the difference between the required and actual value, with the required value being calculated using the derived coefficients:

$$\gamma_{err} = d^{-1}\,[\,-3 \;\; 4 \;\; -1]\,[\,h_g(1)\; h_g(2)\; h_g(3)]^T - \gamma \tag{3-19}$$

$$q_{err} = d^{-2}\,[\,4 \;\; -8 \;\; 4]\,[\,h_g(1)\; h_g(2)\; h_g(3)]^T - q \tag{3-20}$$

$$h_{err} = [\,1 \;\; 0 \;\; 0\,]\,[\,h_g(1)\; h_g(2)\; h_g(3)]^T + 200 - h \tag{3-21}$$

Eqs (3-19), (3-20), and (3-21) can be expressed in matrix form as

$$\begin{bmatrix}\gamma_{err}\\ q_{err}\\ h_{err}\end{bmatrix} = \begin{bmatrix}-3/d & 4/d & -1/d\\ 4/d^2 & -8/d^2 & 4/d^2\\ 1 & 0 & 0\end{bmatrix}\begin{bmatrix}h_g(1)\\ h_g(2)\\ h_g(3)\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 200\end{bmatrix} - \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\gamma\\ q\\ h\end{bmatrix} \tag{3-22}$$

Using Eq (3-22), Eq (3-18) can be rewritten as a matrix that will use the three previous-

ly defined terrain altitudes as inputs and aircraft states as feedbacks:

$$q_{cmd} = [K_\gamma \;\; K_q \; K_h] \begin{bmatrix} -3/d & 4/d & -1/d \\ 4/d^2 & -8/d^2 & 4/d^2 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} h_g(1) \\ h_g(2) \\ h_g(3) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 200 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \gamma \\ q \\ h \end{bmatrix} \qquad (3\text{-}23)$$

A state-space expression for the compensators K_γ, K_q, and K_h can be created using Eqs (3-16) and (3-18) along with the appropriate gains for K_γ and K_h, which were 200 and 15 respectively. The inputs to the state-space will be the error signals that were derived in Eqs (3-19) through (3-22):

$$\begin{bmatrix} \dot{x}_\gamma \\ \dot{x}_h \end{bmatrix} = \begin{bmatrix} -100 & 0 \\ 0 & -100 \end{bmatrix} \begin{bmatrix} x_\gamma \\ x_h \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \gamma_{err} \\ q_{err} \\ h_{err} \end{bmatrix}$$

$$\begin{bmatrix} y_\gamma \\ y_q \\ y_h \end{bmatrix} = \begin{bmatrix} -19640 & 0 \\ 0 & 0 \\ 0 & -1470 \end{bmatrix} \begin{bmatrix} x_\gamma \\ x_h \end{bmatrix} + \begin{bmatrix} 200 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 15 \end{bmatrix} \begin{bmatrix} \gamma_{err} \\ q_{err} \\ h_{err} \end{bmatrix} \qquad (3\text{-}24)$$

The input to the aircraft closed loop plant, q_{cmd}, is equal to the sum of the three outputs from the compensators:

$$\begin{aligned} q_{cmd} &= y_\gamma + y_q + y_h \\ &= [\,-19640 \;\; -1470\,] \begin{bmatrix} x_\gamma \\ x_h \end{bmatrix} + [\; 200 \;\; 1 \;\; 15 \;] \begin{bmatrix} \gamma_{err} \\ q_{err} \\ h_{err} \end{bmatrix} \end{aligned} \qquad (3\text{-}25)$$

where the expressions for the error signals are given by Eq (3-22).

Terrain Model and Evaluation Plan

For this study, the terrain model was represented using the downward-facing portion of a hyperboloid. The equation used to describe the terrain obstacle was

$$z = -(x^2 + y^2)/4000 + 1000 \,;\; 0 \leq z \leq 1000 \qquad (3\text{-}26)$$

where

z = terrain altitude (ft)

x = downrange distance (ft)

y = crossrange distance (ft)

A three-dimensional view of the terrain model is shown in Figure 3.7. Since the evaluation will only be performed flying over the top of the hill, the crossrange distance, y, will always be equal to zero.

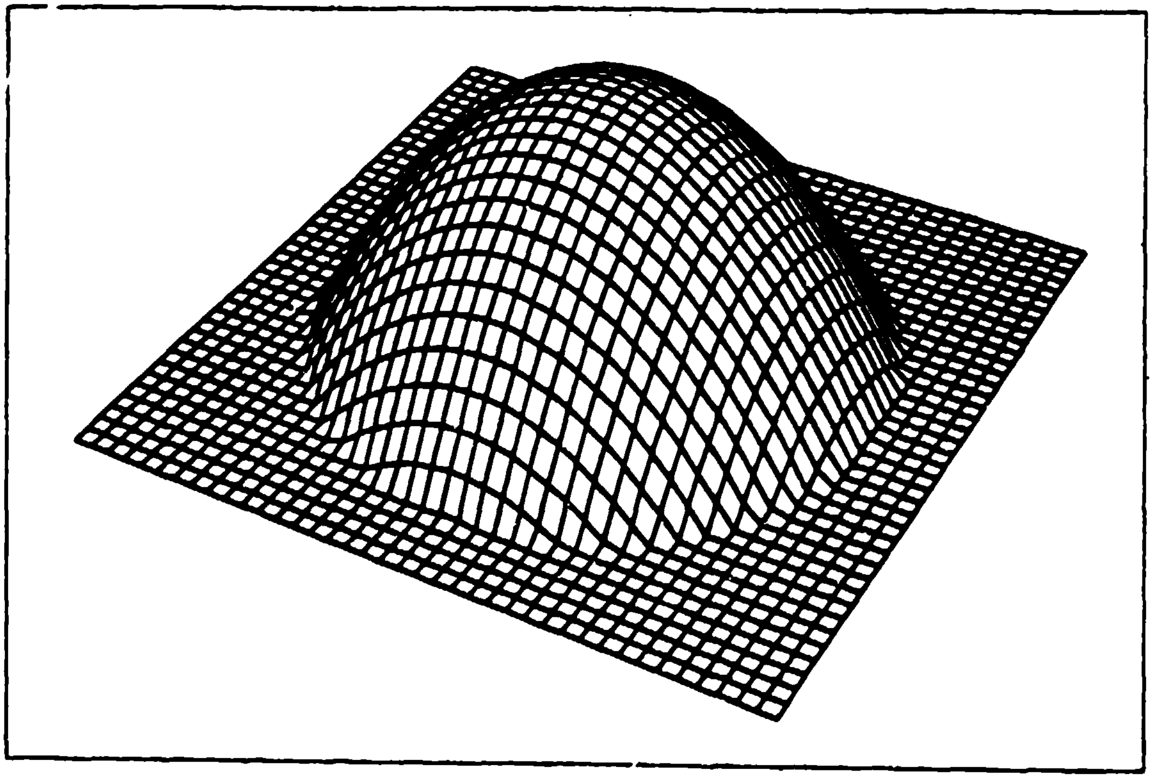


Figure 3.7: Terrain Obstacle Model

Now that the pitch rate input into the closed loop aircraft plant has been expressed in terms of the three terrain altitudes and three state feedbacks, aircraft performance will be evaluated for varying values of look-ahead distance, d. Distances of 0, 300, 600, and 1200 feet will be used to determine if this is a good approach to the terrain avoidance problem. The results, which are addressed in Chapter 4, will be evaluated using plots of aircraft altitude versus ground distance. Digital terrain models will be simulated by biasing the terrain altitude as a function of distance. For example, a terrain model with a look-ahead distance of 300 feet would contain the normal terrain, labeled $h_g(1)$, a second terrain input that is placed 150 feet closer to the aircraft, called $h_g(2)$, and a third terrain input that is placed 300 feet closer to the plane, which is designated as $h_g(3)$. This concept is shown in Figure 3.8 which is an enlarged area of the initial upslope of the hill. Moving the terrain closer to the aircraft is the same as looking farther ahead of the aircraft, therefore, this is the approach that will be used for all look-ahead distances.

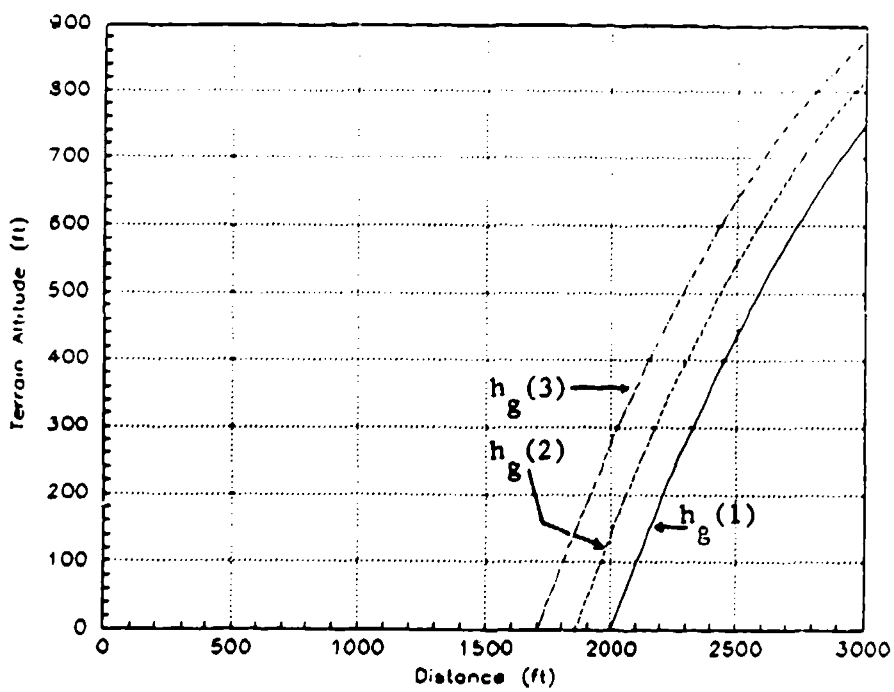


Figure 3.8: Enlarged View of Simulated Terrain Showing the Concept of a 300-foot Look-Ahead Distance

IV. Results and Discussion

Altitude Controller Evaluation

The altitude controller, designed and implemented in Chapter III was evaluated for five values of look-ahead distance: 0 feet, 100 feet, 300 feet, 600 feet, and 1200 feet. Each distance was evaluated against the terrain model which was developed in Chapter III. The evaluation and comparisons made between the various look-ahead distances were based on the altitude response of the aircraft with respect to the terrain.

The first distance evaluated was 0 feet, therefore $h_g(3)$ and $h_g(2)$ were equal to zero. This case is representative of the use of radar altimeters, which essentially look downward from the aircraft to obtain information on terrain altitude. Attack and small fighter aircraft such as the F-16 and A-10 use radar altimeters for this purpose. As can be seen in Figure 4.1, the aircraft did not avoid the terrain due to the sharp rise. This is similar to using a radar altimeter, not including the altimeter cone model, for terrain avoidance. Over gentle terrain, the radar altimeter will work well as a sensor because the lag time between sensing of the terrain and aircraft response is small compared to the rate at which the terrain rises, thus providing the aircraft with ample time to respond. Even though the aircraft had problems negotiating the initial terrain rise, it did reach the desired peak value of 1200 feet MSL, or 200 feet above the terrain peak and followed the backside of the hill rather well. Using Figure 4.1, the lag time for aircraft response can be measured as approximately 0.5 seconds which corresponds with the rise time that was observed in Chapter III for a step input.

The next distance evaluated was 100 feet, which corresponds to the value of $h_g(3)$; $h_g(2)$ took on a distance of 50 feet for this case. All three loops of the altitude controller will have pitch rate inputs. Figure 4.2 shows the results of this test distance. Again, the F-16 crashed into the terrain obstacle, but a very slight improvement in

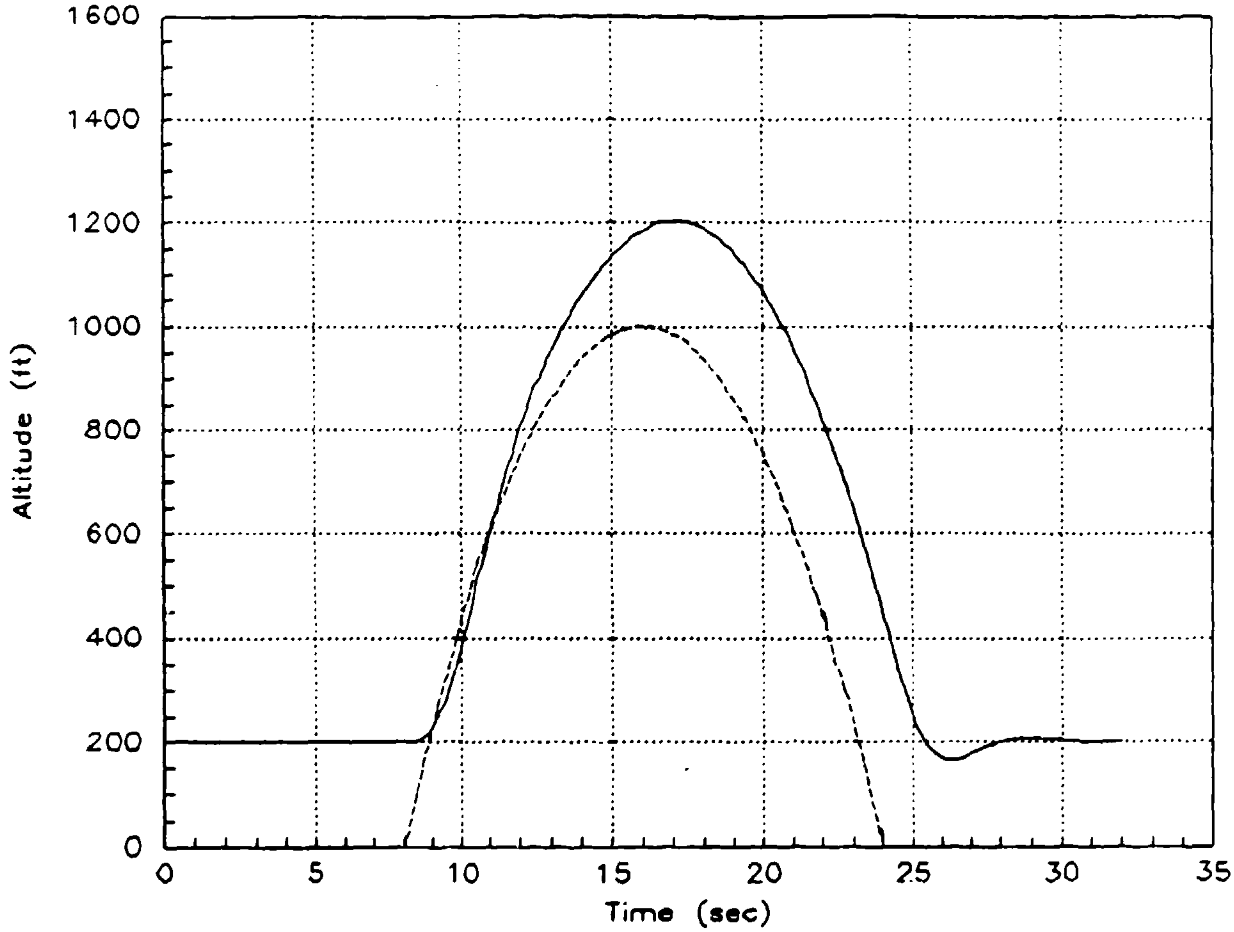


Figure 4.1: Altitude Response vs Terrain for 0-foot Look-Ahead Distance

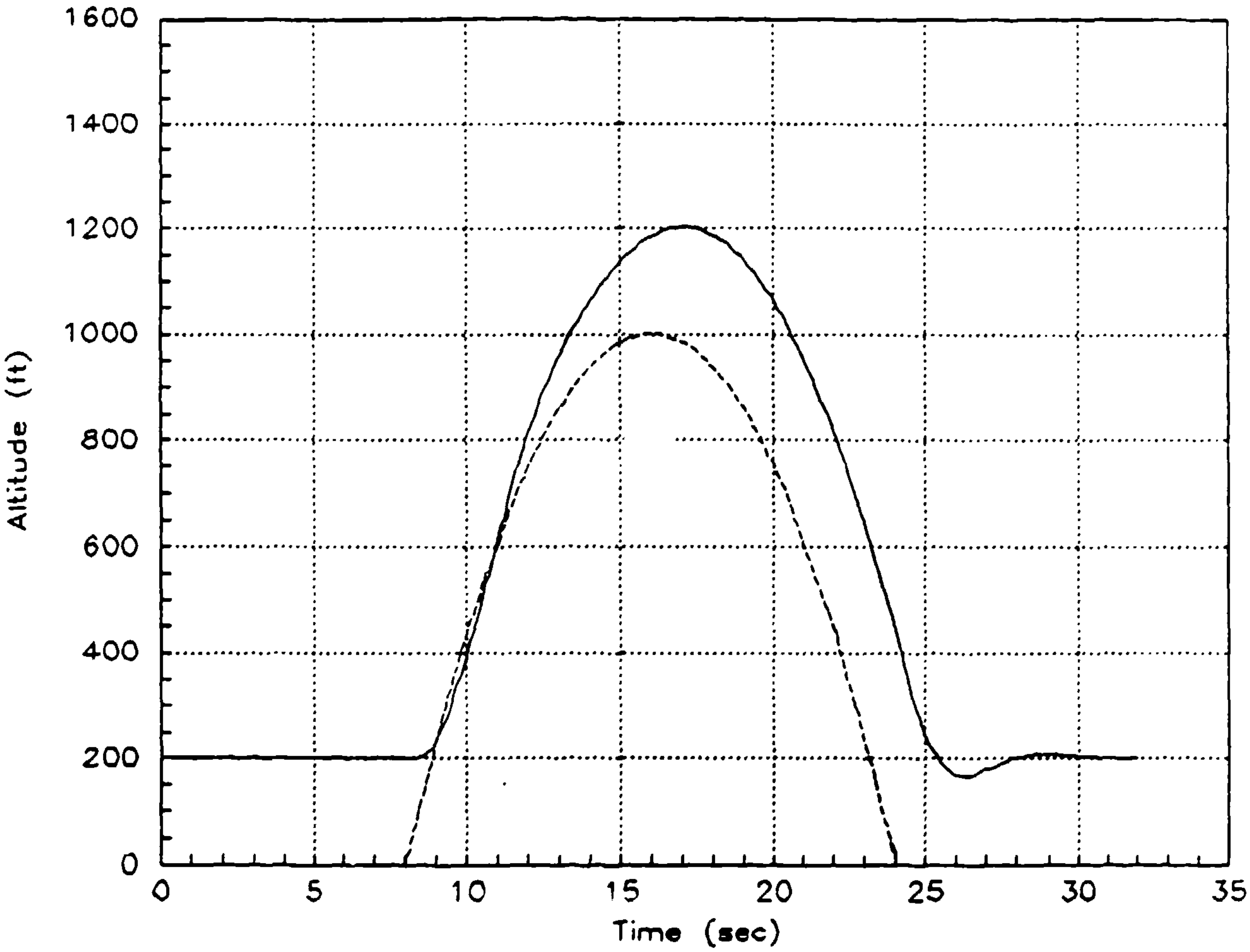


Figure 4.2: Altitude Response vs Terrain for 100-foot Look-Ahead Distance

response can be seen. Referring back to the flight path angle response in Figure 3.5, one can see that the flight path loop of the controller would not have ample time to build up a significant input value. A 100 foot look-ahead distance for an aircraft travelling at 670 feet per second only corresponds to an additional 0.15 seconds of response time. Therefore, not much improvement could be expected for this case.

The next look-ahead distance evaluated was 300 feet. Although the aircraft still penetrated the terrain model slightly, it did show a significant improvement over the previous two cases. Figure 4.3 illustrates these results. The initial response of the aircraft occurred approximately 0.5 seconds prior to when the altitude loop began feeding inputs into the system which should be expected for a 300 foot look-ahead distance. However, the initial response was in the wrong direction due to the nonminimum phase nature of the flight path angle loop. Still, the overall response was an improvement in comparison to the 0 and 100 foot cases.

The nonminimum phase response of the flight path angle loop was more pronounced for a distance of 600 feet since there was twice as much time available, compared to the 300 foot case, before the altitude loop commanded inputs. As shown in Figure 4.4, the F-16 just barely avoided the terrain due to the larger look-ahead distance. The nonminimum portion of the flight path angle response subsided approximately 0.5 seconds before the aircraft reached the beginning of the terrain obstacle, giving the aircraft a slight amount of positive pitch rate.

As with the all of the previous three cases, the aircraft reached a maximum altitude of 1200 feet, or 200 feet above the terrain, as was desired with the peak altitude occurring closer to the peak of the terrain. This indicates that the implementation scheme is working as intended since information about the upcoming terrain is obviously being used in the calculation of the pitch rate command input.

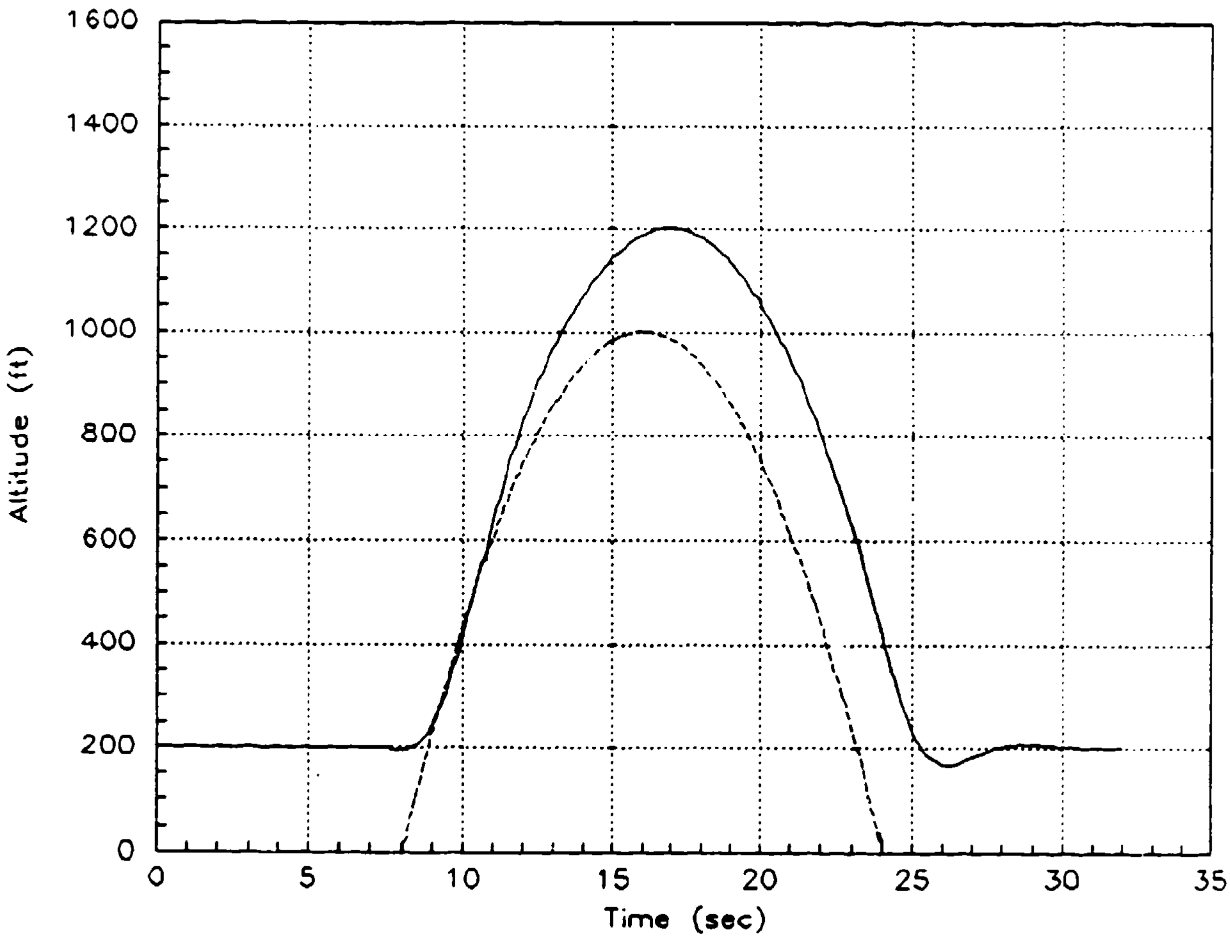


Figure 4.3: Altitude Response vs Terrain for 300-foot Look-Ahead Distance

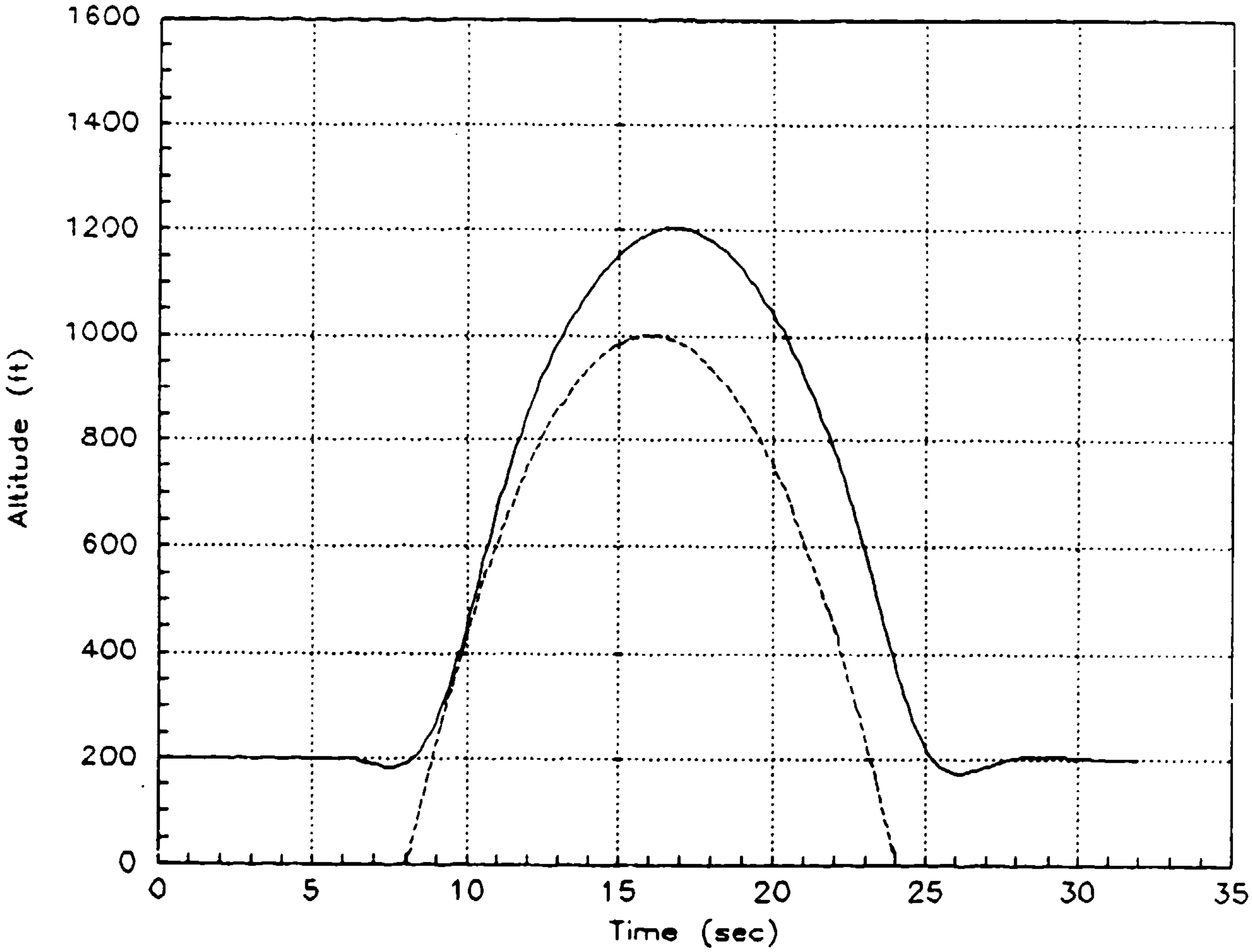


Figure 4.4: Altitude Response vs Terrain for 600-foot Look-Ahead Distance

The final look-ahead distance evaluated in this thesis was 1200 feet. Dramatic improvements in aircraft altitude response were evident as can be seen in Figure 4.5. The F-16 nearly followed the entire terrain obstacle using the 1200 foot distance. Although the portion of nonminimum phase response is slightly longer, the amount of pitch rate built-up by the time the aircraft reached the beginning of the terrain negated the rise time delay for the altitude loop that was seen in the earlier cases. Also note that the larger look-ahead distance decreased the overshoot of the 200 foot target altitude at the end of the terrain avoidance maneuver.

In order to achieve a better feeling for the spatial relationship between the aircraft and the terrain, Figure 4.6 has been included to show aircraft altitude as a function of downrange distance from the initiation point of the test run. The aircraft required approximately one mile of distance to fly over the 1000 foot high hill.

To confirm that the flight path angle was the cause of the initial nonminimum phase response of the aircraft, a test case was run with all of the gains in the flight path loop set to zero: in other words, only the altitude and pitch rate loops of the controller were providing pitch rate inputs into the aircraft flight control system. As suspected, the terrain avoidance performance of the F-16 degraded significantly in the absence of flight path controller loop inputs, which can be seen in Figure 4.7. The performance of the aircraft with a 1200 foot look-ahead distance is very similar to the 100 foot case, and this indicates that the pitch rate inputs are insignificant at longer distances. Referring back to Figure 3.2, the reason pitch rate inputs become insignificant at large look-ahead distances is due to the $1/d^2$ term that is present after the summing junction for the altitudes in the pitch rate loop. Therefore, it can be postulated that a first order equation probably would have performed just as well as the second order one used in this study.

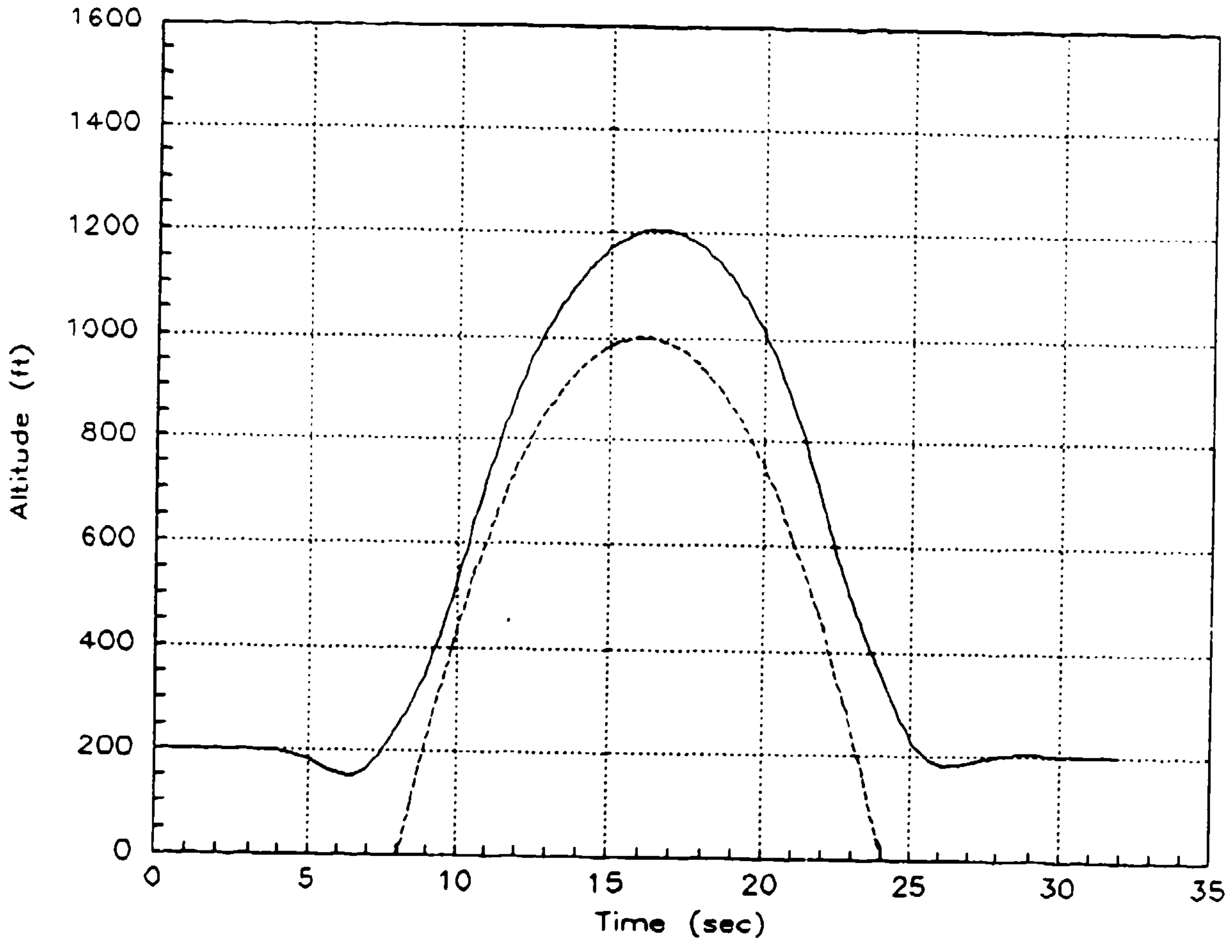


Figure 4.5: Altitude Response vs Terrain for 1200-foot Look-Ahead Distance

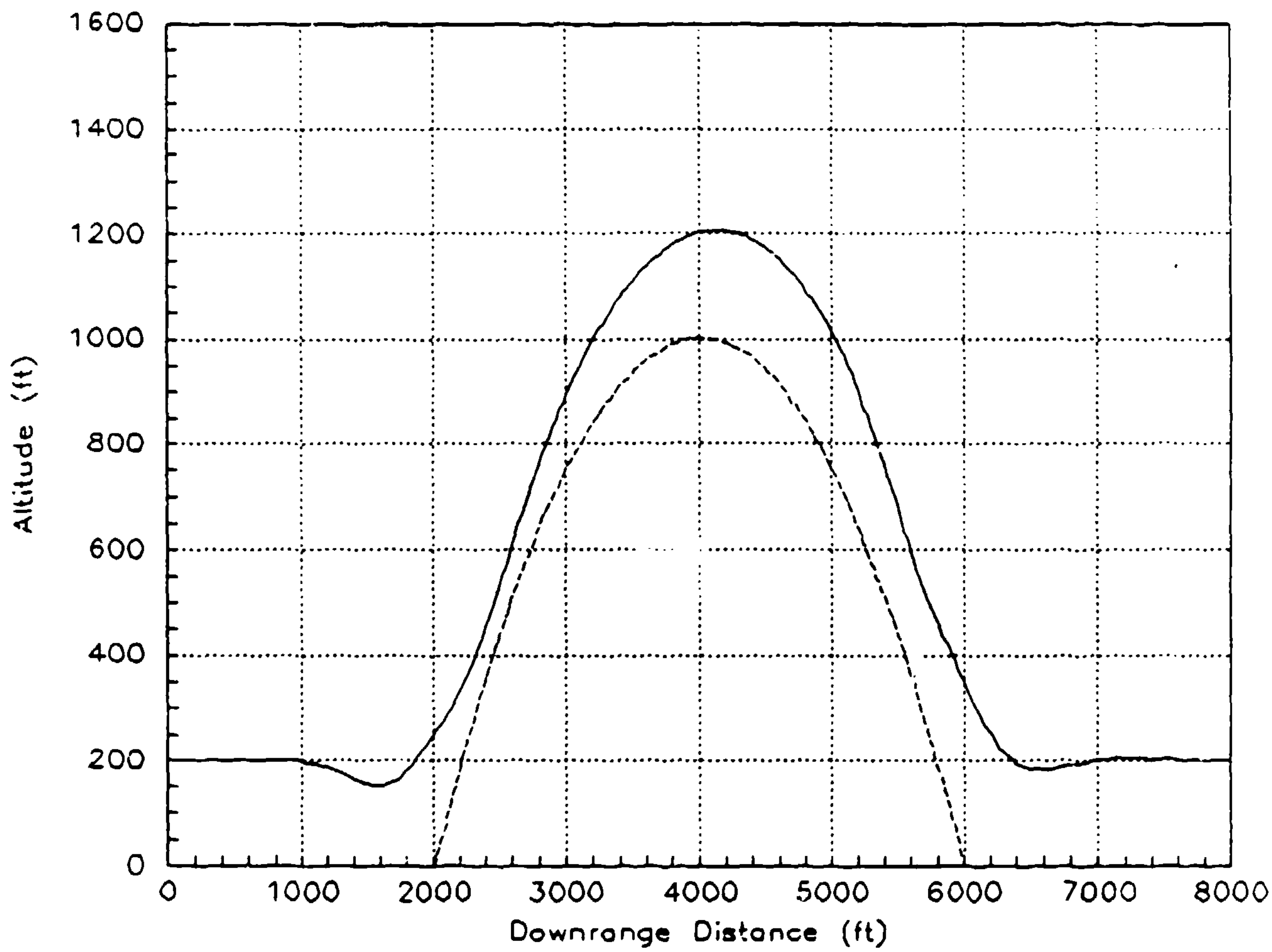


Figure 4.6: Altitude vs Range for 1200-foot Look-Ahead Distance

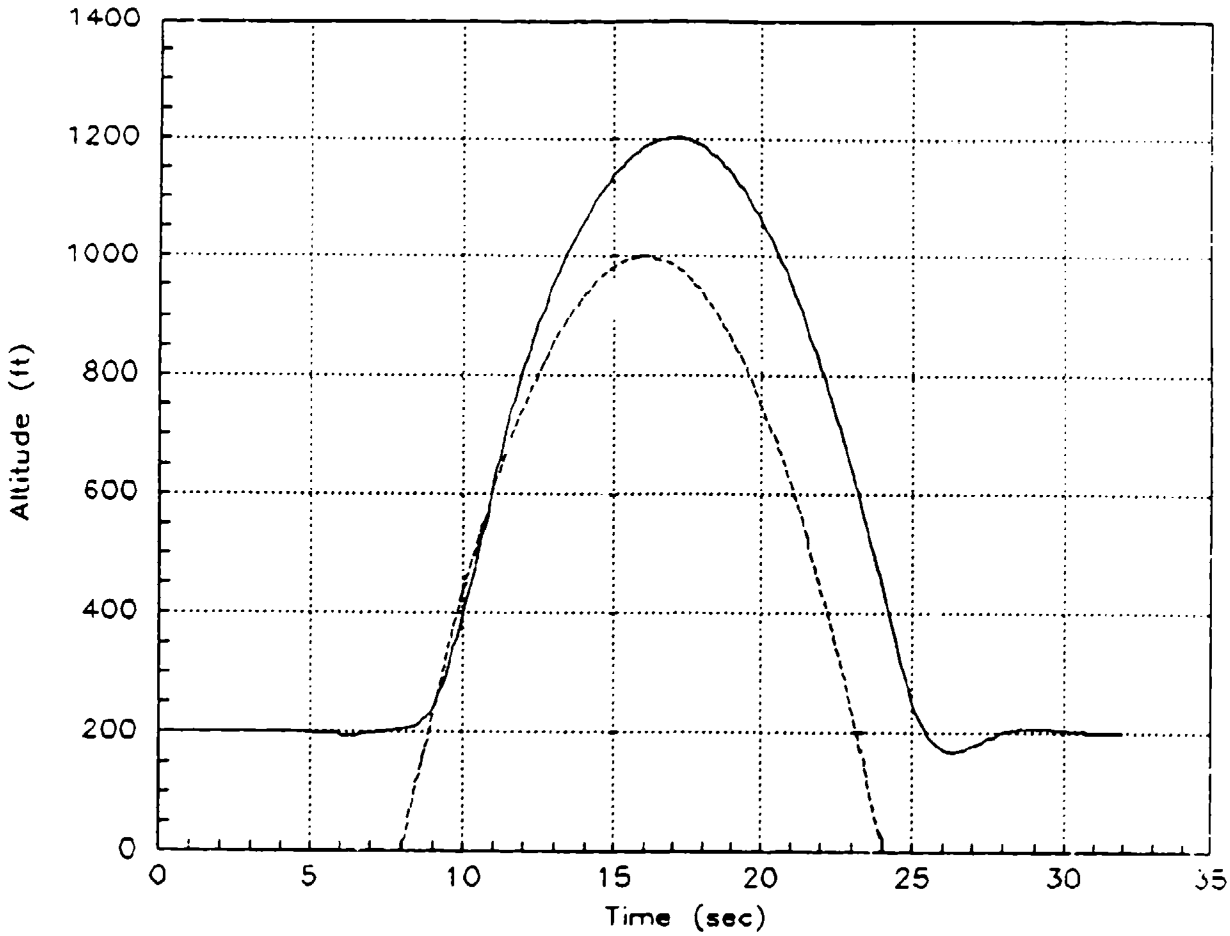


Figure 4.7: Altitude Response With Reduced Gain in Flight Path Loop for 1200-foot Look-Ahead Distance

One of the constraints placed on the terrain avoidance problem in this study was that the aircraft could not exceed $\pm 4g$ of incremental load factor or $-3g \leq A_n \leq 5g$. As seen in Figure 4.8, incremental load factor, represented by the dashed curve, reached a maximum value of approximately 2.7g which corresponds to an actual load factor of 3.7g. Aircraft pitch rate response is also shown in Figure 4.8 along with altitude response versus terrain; the altitude response is shown as a reference for correlation purposes. Figure 4.9 contains the time response plot for horizontal tail deflection during the terrain avoidance maneuver. As can be seen, the deflections did not exceed the limits of ± 25 degrees, and reached a maximum value of almost 8 degrees.

Figure 4.10 contains a comparison summary of altitude error for each of the five look-ahead distances evaluated. Note that the altitude error becomes smaller as the look-ahead distance is increased, which is what was desired. The line corresponding to -200 feet of altitude error represents the terrain, therefore, any curve falling below that line indicates that the aircraft impacted the terrain. While the 1200 foot look-ahead distance does show a significant improvement over the other distances evaluated, it still has wide variations in altitude error (-105 feet to 140 feet). For this reason, an experimental test case was carried out using an approach that was slightly modified from the one presented in this subsection.

Alternate Terrain Avoidance Approach and Evaluation

An alternate approach was tried for implementing a terrain avoidance system to see if any improvements could be made to the altitude response of the aircraft. Referring back to Figure 3.6 which shows aircraft response to a step altitude input, one can see that it takes approximately 0.45 seconds for the aircraft to reach the value of the commanded input. This lag time roughly corresponds to a distance of 300 feet given a velocity of 670 feet per second. Therefore, if the aircraft receives terrain information

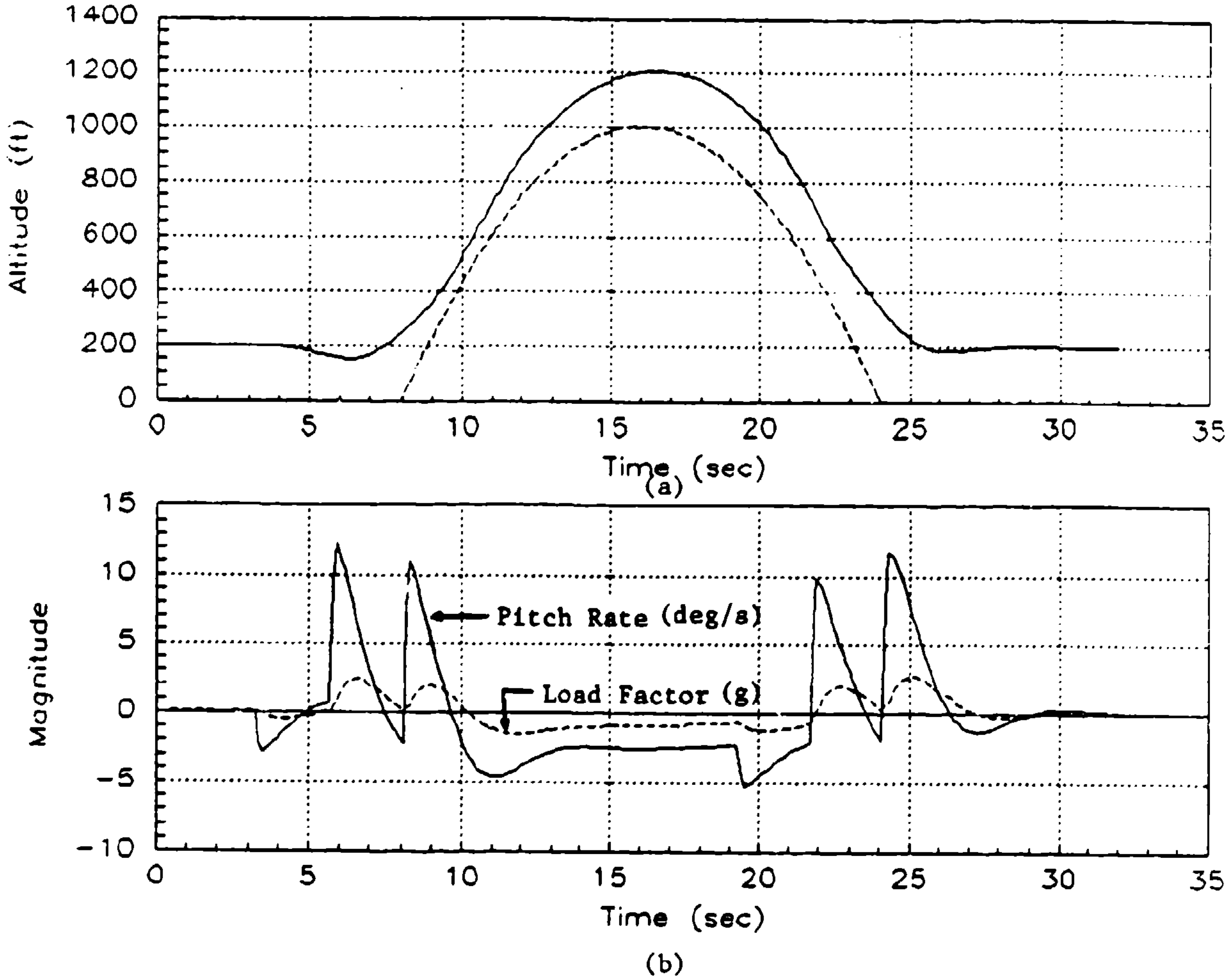


Figure 4.8: Aircraft Pitch Rate and Load Factor Response for 1200-foot Look-Ahead Distance: (a) Altitude vs Terrain, (b) Pitch Rate and Load Factor Response

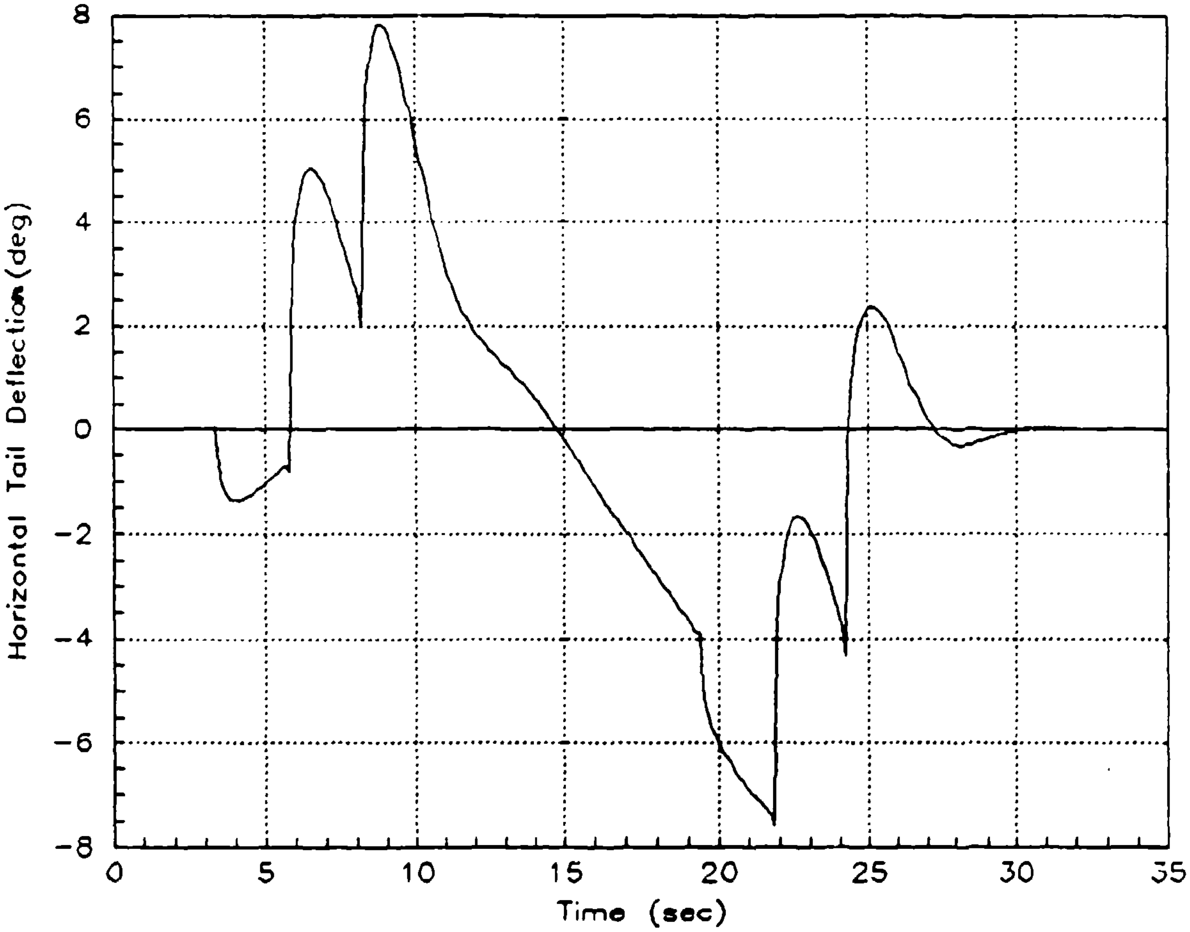


Figure 4.9: Horizontal Tail Response for Terrain Avoidance Maneuver With 1200-foot Look-Ahead Distance

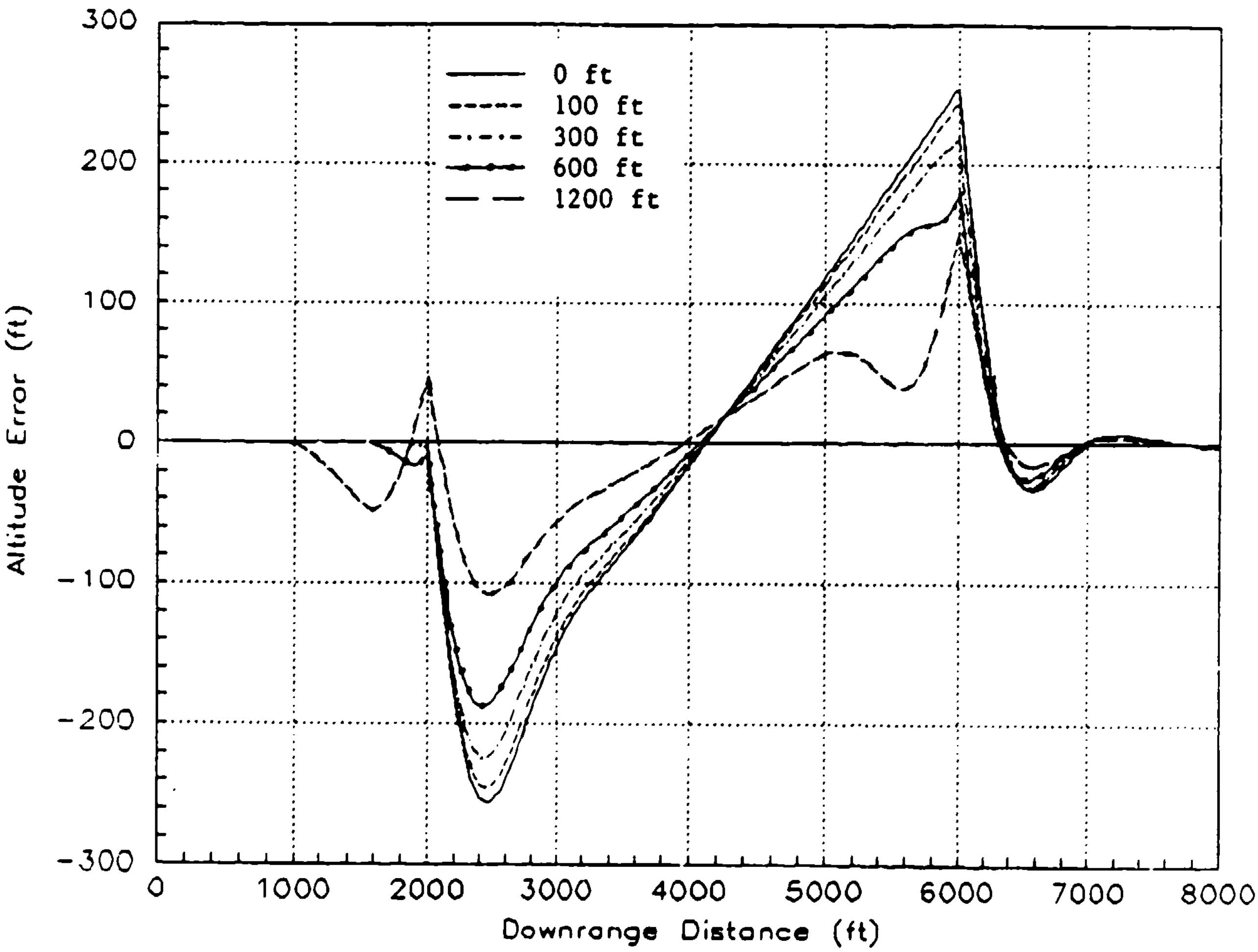


Figure 4.10: Altitude Error vs Range for Various Look-Ahead Distances

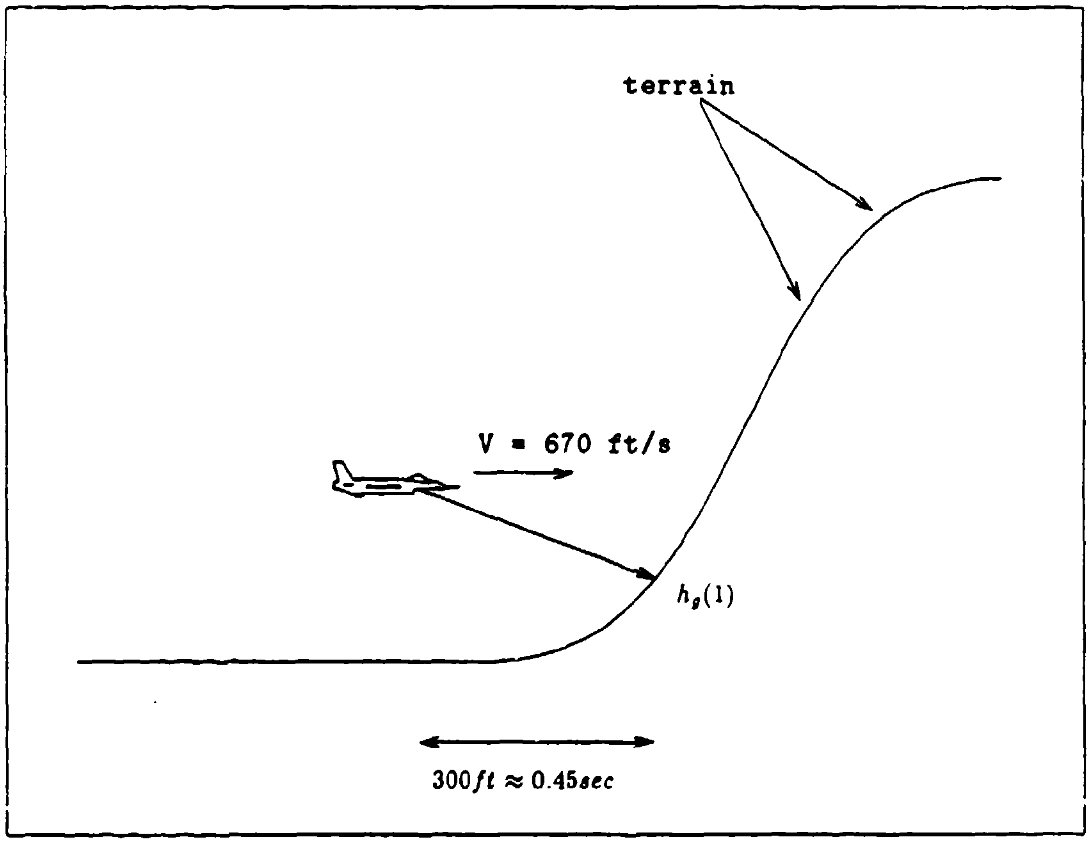


Figure 4.11: Alternate Method for Implementing Terrain Avoidance

300 feet in advance, it should be at that terrain altitude plus 200 feet by the time it actually arrives at that point in space. This approach is illustrated in Figure 4.11. If the initial conditions of the aircraft were different, all that would need to be changed is the look-ahead distance. For normal operating airspeeds, look-ahead distance would be proportional to horizontal velocity. At slower airspeeds, this distance would need to be increased since load factor capabilities degrade in this operating regime.

In implementing this approach, the system which had already been developed can be used with a few minor modifications. The look-ahead altitudes referred to as $h_g(2)$ and $h_g(3)$ in the previous section will now be set to zero, as will the $h_g(1)$ input to the pitch rate and flight path angle loops of the controller. The only point in the loop where $h_g(1)$ will be input is in the altitude loop of the controller. What is actually being done is to make the aircraft think that the terrain lying 300 feet ahead actually lies below. The time response for aircraft altitude versus terrain altitude is shown in Figure 4.12. Comparing Figures 4.12 and 4.5, several conclusions can immediately be drawn. First, the total time required to traverse the terrain is about two seconds less using this approach. Second, the nonminimum phase response of the aircraft is eliminated since flight path angle is no longer commanded which results in a quicker overall response. Third, peak aircraft altitude occurs closer to peak terrain altitude using this modified approach. The results from implementing this approach show that the aircraft altitude response produced less altitude error compared to the error produced using the 1200 foot look-ahead distance, as is shown in Figure 4.13. Altitude error remains within about ± 30 feet using the modified system compared to the 1200 foot look-ahead distance error which ranges between -105 feet and +150 feet.

Examining the pitch rate and incremental load factor response, seen in Figure 4.14 indicates that actual load factor momentarily exceeds the 5g limit by reaching 5.5g. This is a much more aggressive response compared to the response seen using the 1200

foot look-ahead distance, which accounts for the decreased amount of time required to traverse the hill. The increased response can probably be attributed to the fact that no commanded inputs are coming from the flight path angle or pitch rate loops of the controller. Using a 1200 foot look-ahead distance, these two loops will begin commanding negative values of pitch rate while the aircraft is still climbing up the front side of the terrain, thus decreasing the overall commanded pitch rate and resultant load factor. However, they do have a distinct advantage during the initial response to a terrain obstacle.

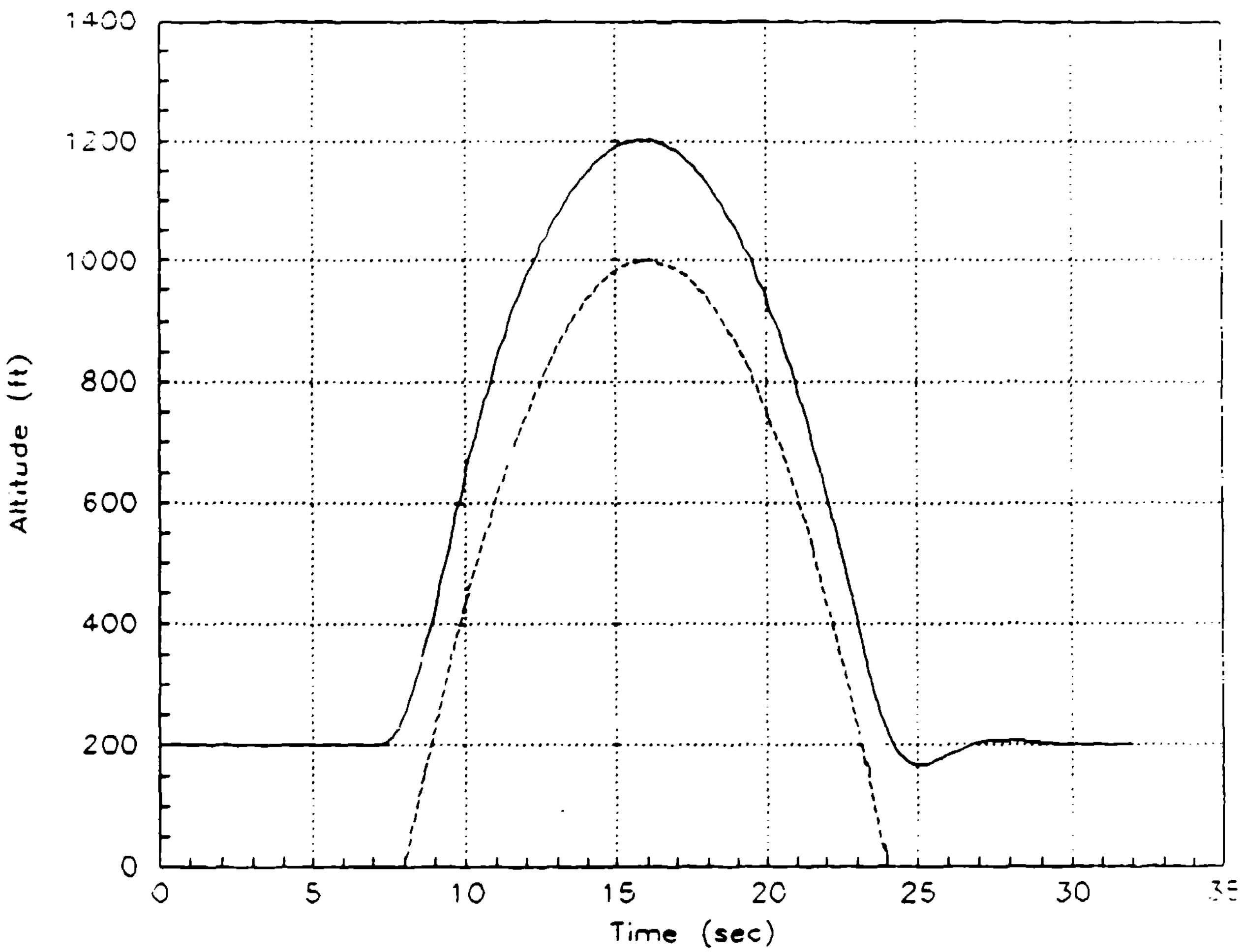


Figure 4.12: Aircraft Response Using Modified Approach for 300-foot Look-Ahead Distance

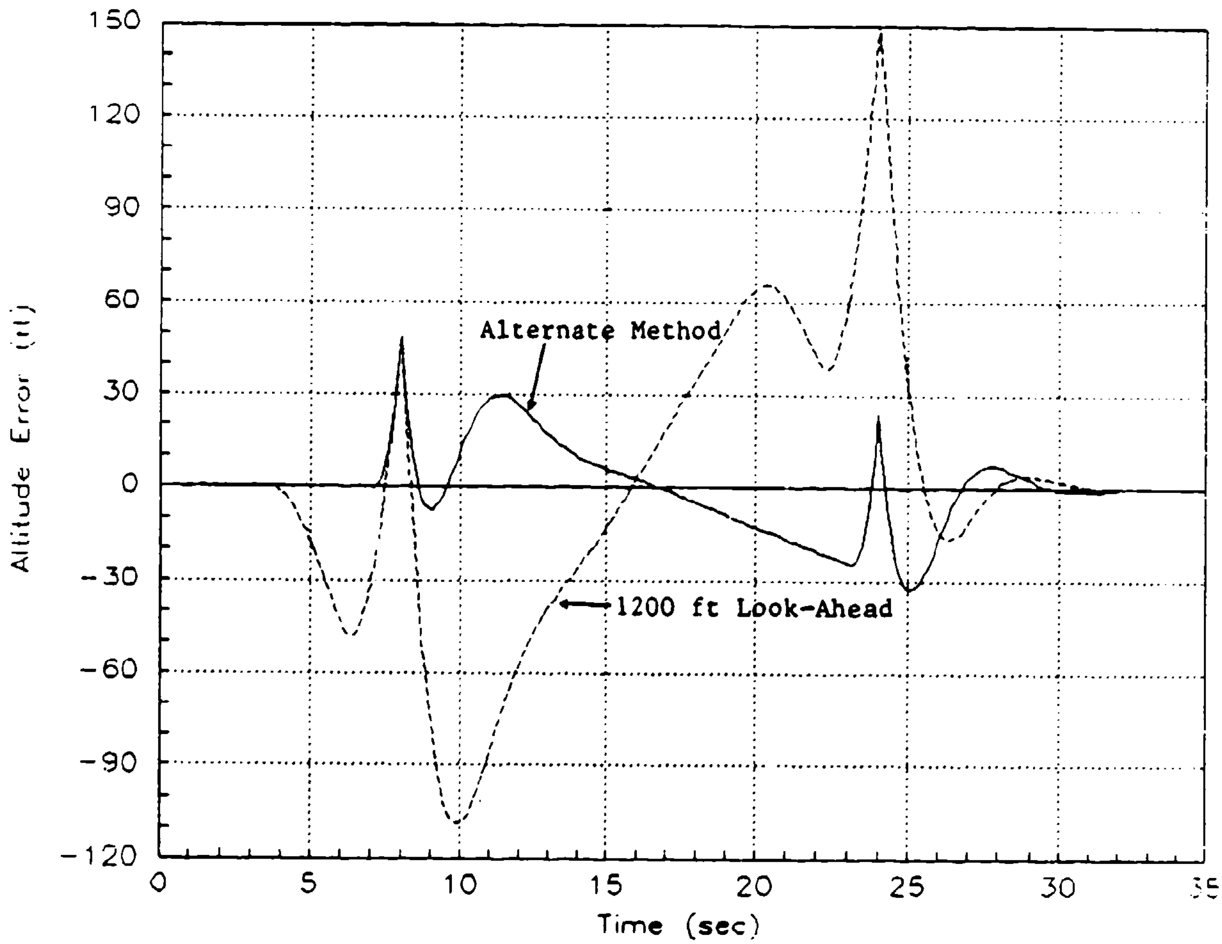


Figure 4.13: Altitude Error Comparison Between Different Terrain Avoidance Implementations

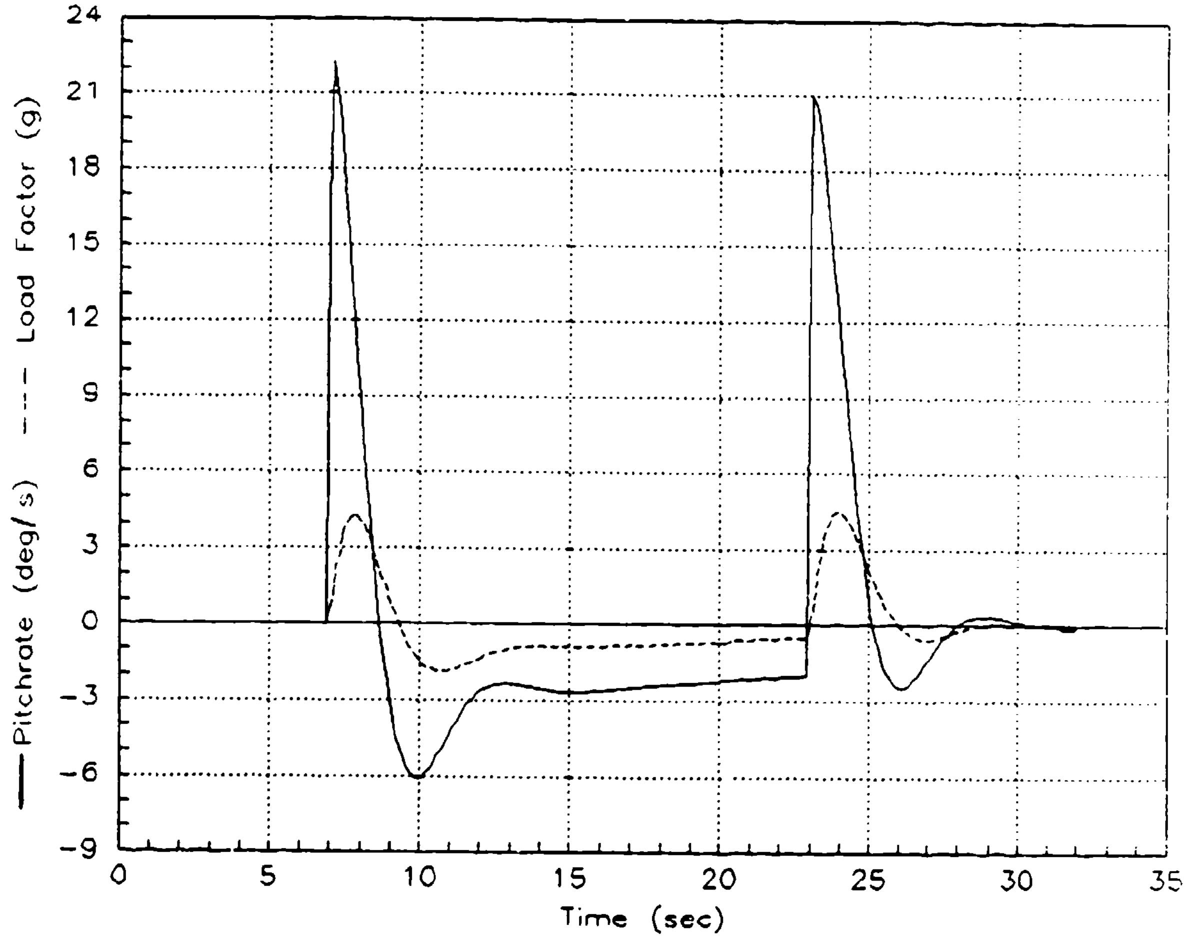


Figure 4.14: Aircraft Pitch Rate and Load Factor Response for Modified Terrain Avoidance Approach

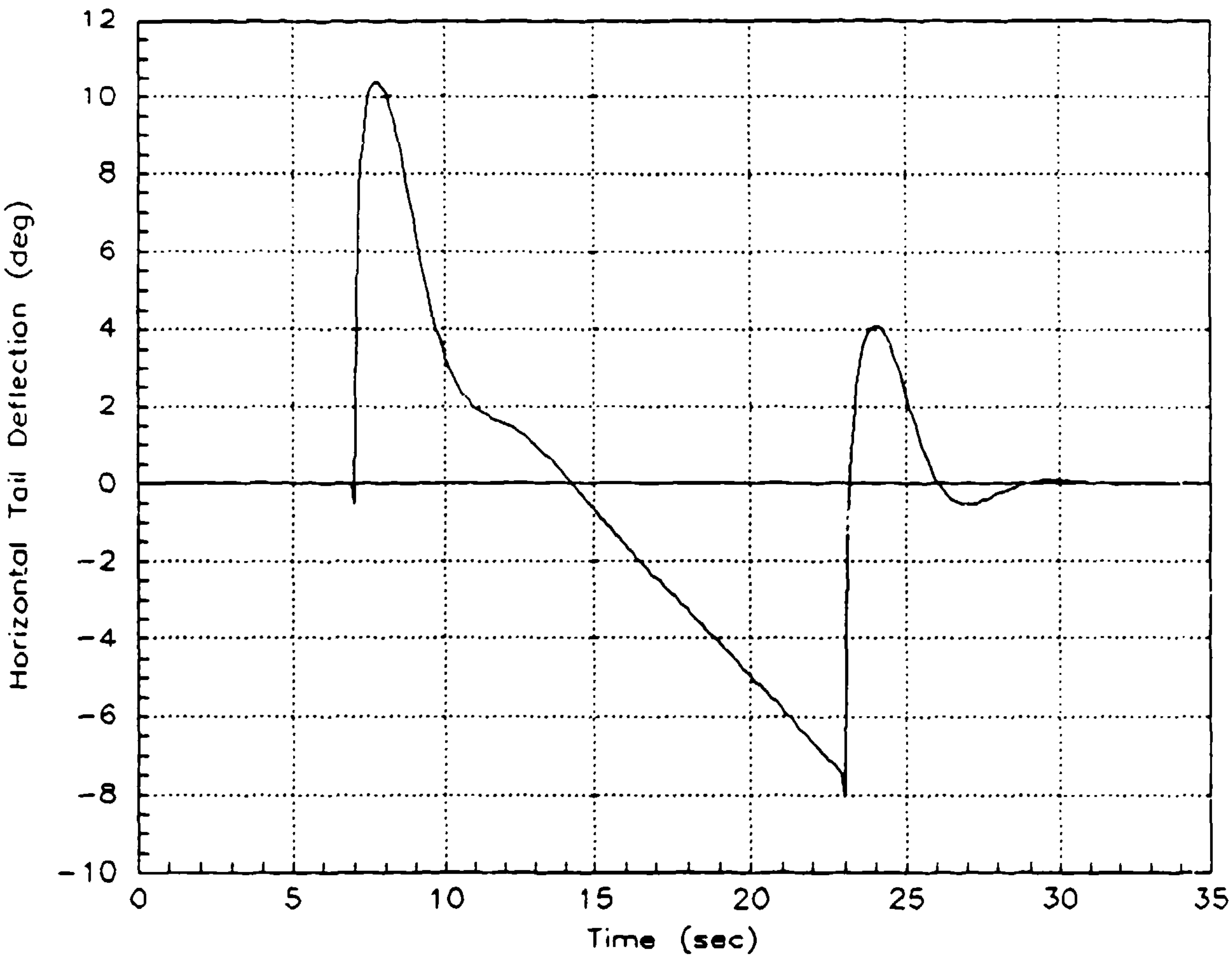


Figure 4.15: Horizontal Tail Response for Modified Terrain Avoidance Approach

V. Conclusions

Based on the results presented in Chapter IV, some conclusions can be reached about the effectiveness of each of the two terrain avoidance schemes: one based on three different reference altitudes using a look-ahead distance, and the other simply based on information about the terrain lying 300 feet ahead of the aircraft. Some conclusions can also be drawn about the potential of the digital terrain database with respect to ground collision avoidance systems.

Using the altitude controller that was derived in Chapter III and evaluated in Chapter IV, a minimum look-ahead distance was required in order for the aircraft to effectively avoid the terrain. A look-ahead distance of 600 feet provided enough advance terrain altitude information for the aircraft to just avoid terrain impact. Using the original approach of utilizing three different terrain altitudes, the only look-ahead distance evaluated that provided sufficient altitude separation between the aircraft and the terrain was 1200 feet. Even then, the altitude error had a variation of 255 feet between the minimum and maximum error values. The design did return the aircraft to the initial conditions of level flight and 200 feet in altitude after traversing the terrain obstacle, as was required. The alternate approach of looking a set distance ahead of the aircraft provided a better terrain avoidance capability.

A defect in the design of the terrain avoidance system was the response of flight path angle in the altitude control loop. While it was a good idea in theory for the purpose of aligning the aircraft velocity vector with the slope of the terrain, the flight path angle loop of the controller exhibited a nonminimum phase response which gave the controller some drawbacks. However, the flight path angle loop greatly enhanced aircraft response when larger look-ahead distances such as 600 and 1200 feet were used. For shorter distances, this loop was ineffective due to a somewhat sluggish response.

An attempt was made to use pitch angle, θ, instead of flight path angle for following the slope of the terrain, but this design proved ineffective because a change in pitch angle did not produce an equivalent change in flight path angle. The result was aircraft impact into the terrain. The pitch rate loop was ineffective for any reasonable look-ahead distance because of the squared distance term in the denominator of the forward path gain.

All of these factors, when combined, resulted in very little improvement in the terrain avoidance capabilities of the F-16 for look-ahead distances less than 600 feet. A redesign of the flight path angle and pitch rate loops of the controller could result in better response characteristics for the controller, however, it is questionable if the overall performance of the terrain avoidance system would improve. Satisfactory performance could be achieved for distances greater than 1200 feet.

The performance of the alternate terrain avoidance implementation showed a dramatic improvement in the capabilities over the system that was just discussed. The altitude errors of the terrain avoidance system were reduced to $\pm$30 feet by converting the rise time of the altitude control loop from seconds to a distance and moving the terrain reference point this distance out in front of the aircraft. This sort of implementation was a more intuitive approach to implementing a ground collision avoidance system. This design also returned the aircraft to the initial conditions of level flight and 200 feet in altitude after traversing the terrain obstacle, as was required.

The terrain avoidance system that was designed for this study was based on only one condition. For other flight conditions, or off design cases, the required look-ahead distance will change. Look-ahead distance should be increased for faster airspeeds and decreased for slower airspeeds up to a certain point. At flight conditions where the maximum allowable load factor cannot be achieved, which was 5g for this study, the look-ahead distance will need to be increased in order to allow the for the slower

response time of the aircraft at the reduced maximum achievable load factor. The problem of implementing a variable look-ahead distance could be accomplished using a schedule similar to the variable control system gains that are based on impact pressure.

Both of the GCAS implementations discussed in this thesis made use of potential of the digital terrain database (DTD). Obtaining terrain information at a series of distances in front of the aircraft is a task that is tailored to the capabilities of the DTD. Using the DTD, this distance could be varied according to flight conditions. In addition, information on the surrounding terrain could also be obtained at the same time without the use of a dedicated sensor meaning a GCAS could be designed to maneuver in the lateral-directional plane. The DTD will most likely be an integral part of any future terrain avoidance system.

VI. Recommendations

During the course of this thesis, several additional areas of interest have emerged which should be evaluated. Each of these areas of interest have the potential for advancing the solution of the terrain avoidance problem. These areas are as follows:

1. Implementation of the two terrain avoidance systems developed in this thesis into an F-16 simulator so that a more detailed study can be conducted on the effects of look-ahead distance on terrain avoidance capabilities. An investigation could also be done using ,/brid system that combines the features of both systems used in this thesis.
2. Developm*nt of a three degree of freedom ground collision avoidance system which can maneuver in the lateral-directional axis in order to avoid terrain.
3. Develoment of a terrain avoidance system using optimal control theory for determining the path for minimum distance or for minimum time around a terrain obstacle.

The first recommendation is required in order to validate the results of this thesis. A study should be done to determine the effects of look-ahead distance on terrain avoidance capabilities and what the required minimum distance is. Different terrain obstacl and slopes should also be used in order to determine their effects on terrain avoidance performance. Several F-16 simulations are available at the Flight Dynamics Laboratory and can be connected to terrain boards or digital terrain databases. The simulations are written in FORTRAN computer code and would require modification

in order to incorporate the altitude controller developed in this thesis. Using results from the simulator, a comparison could be made to ascertain the potential of the terrain avoidance systems developed in this thesis.

The second recommendation was made because no current, automatic GCAS design incorporates maneuvering in the lateral-directional axis. This is an area that has considerable potential in the tactical combat arena since maneuvering in the longitudinal axis can often increase aircraft exposure time to enemy defenses. Maneuvering in the lateral-directional axis could have the potential of using terrain obstacles to mask the aircraft from enemy radar. In addition, this could enhance the terrain avoidance performance of the aircraft in mountainous terrain where vertical pull-up maneuvers may not be effective at low altitude.

If this recommendation were pursued, some sort of bank angle hold loop would be required since, as shown in Chapter 2, the F-16 state-space model could not maintain a non-zero bank angle. An alternate approach to this problem could be the addition of a lag compensator that would force the spiral mode root closer to the imaginary axis, thus slowing down the effects of this root on roll rate performance. Presumably, if this problem were overcome, then a variation of the altitude controller developed in this thesis could be used since the look-ahead distance vector would be translated through the pitch angle and bank angle.

The third recommendation involves a complex area of control theory. Using the terrain obstacle developed in this thesis, optimal control theory should be able to define the minimum time or minimum distance path around the terrain along with the optimal control law for the pitchrate and roll rate inputs required to fly this path.

Using pseudo 'bang-bang' control for pitchrate inputs of ±11 degrees per second, a time history of aircraft altitude versus terrain altitude was generated. This time history and the time history of the other aircraft states are shown in Figures 6.1 through 6.4.

Figure 6.1 shows that the time required to traverse the terrain obstacle is slightly less that 14 seconds compared to a time of approximately 18 seconds from the system evaluated in this thesis. Since Figure 6.1 is using the maximum pitchrate authority limits, this should be close to what optimal control theory would predict.

Another approach using optimal control would involve designing the altitude controller using loop transfer recovery techniques. This involves setting up a linear quadratic cost function with weightings on the aircraft states and penalties on the controls. This approach is used for the infinite horizon problem where time is not a constraint, and the solution to the optimal control problem is based on the solution of the Ricatti equation.

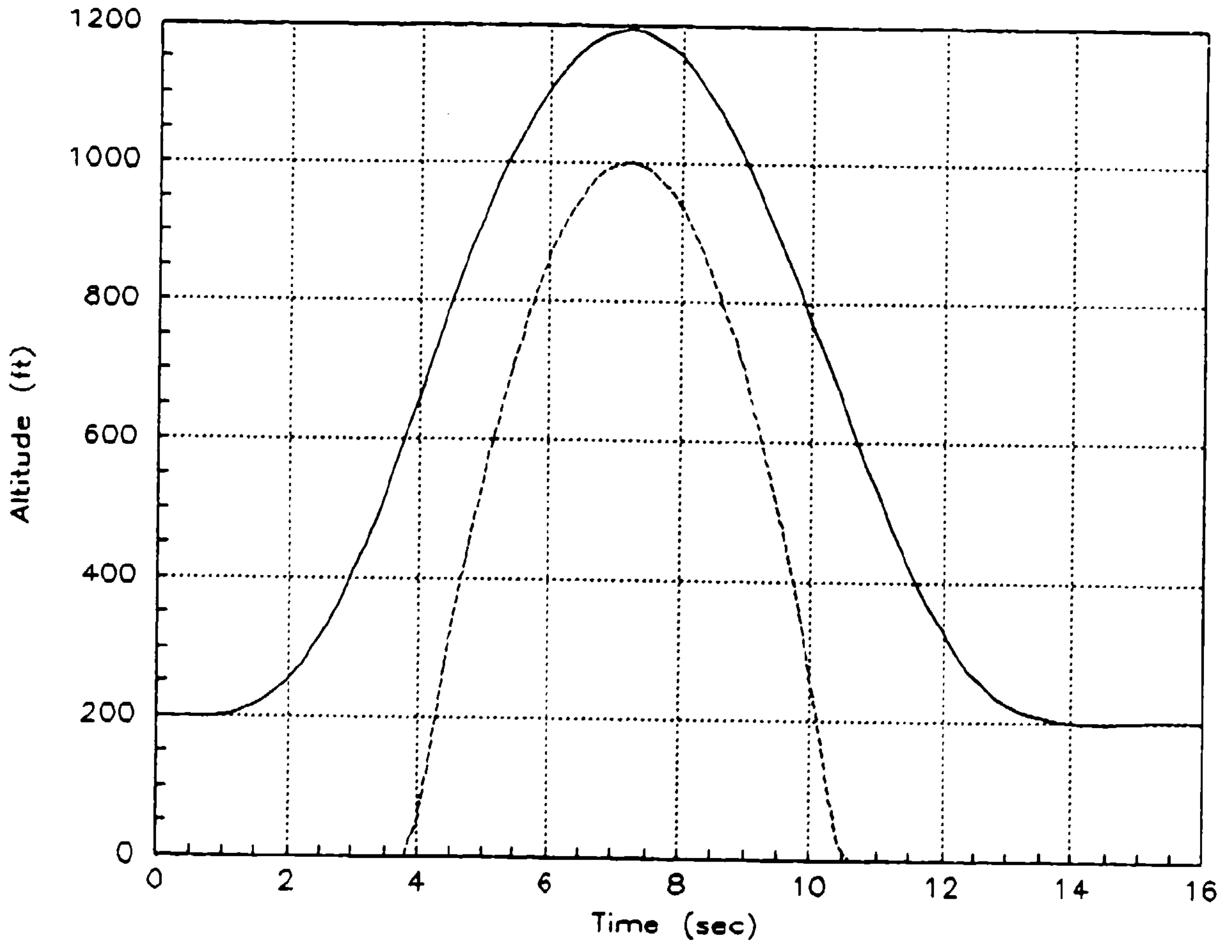


Figure 6.1: Time History of F-16 Terrain Avoidance
Using 'Bang-Bang' Inputs

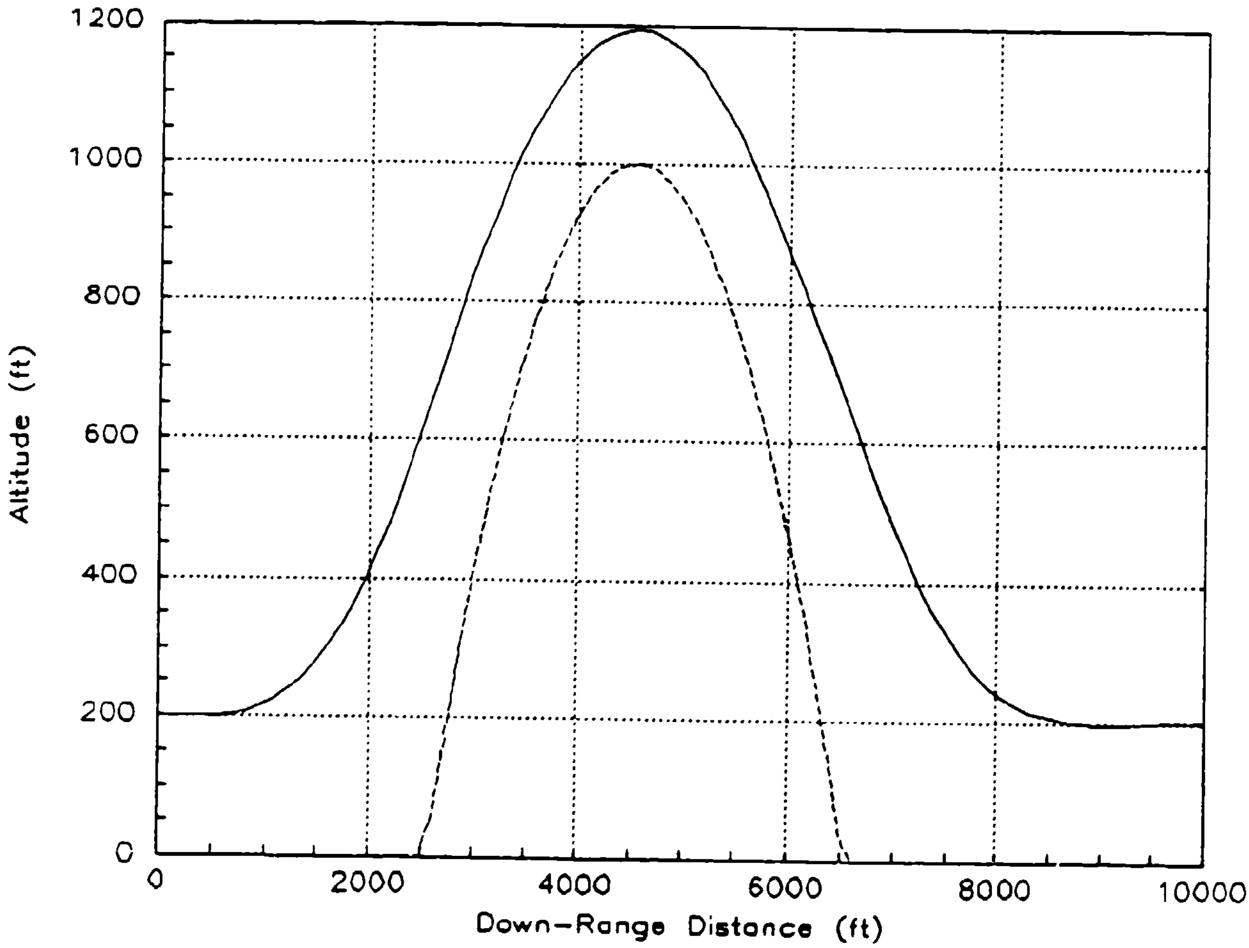


Figure 6.2: Aircraft Altitude vs Range Using 'Bang-Bang' Inputs

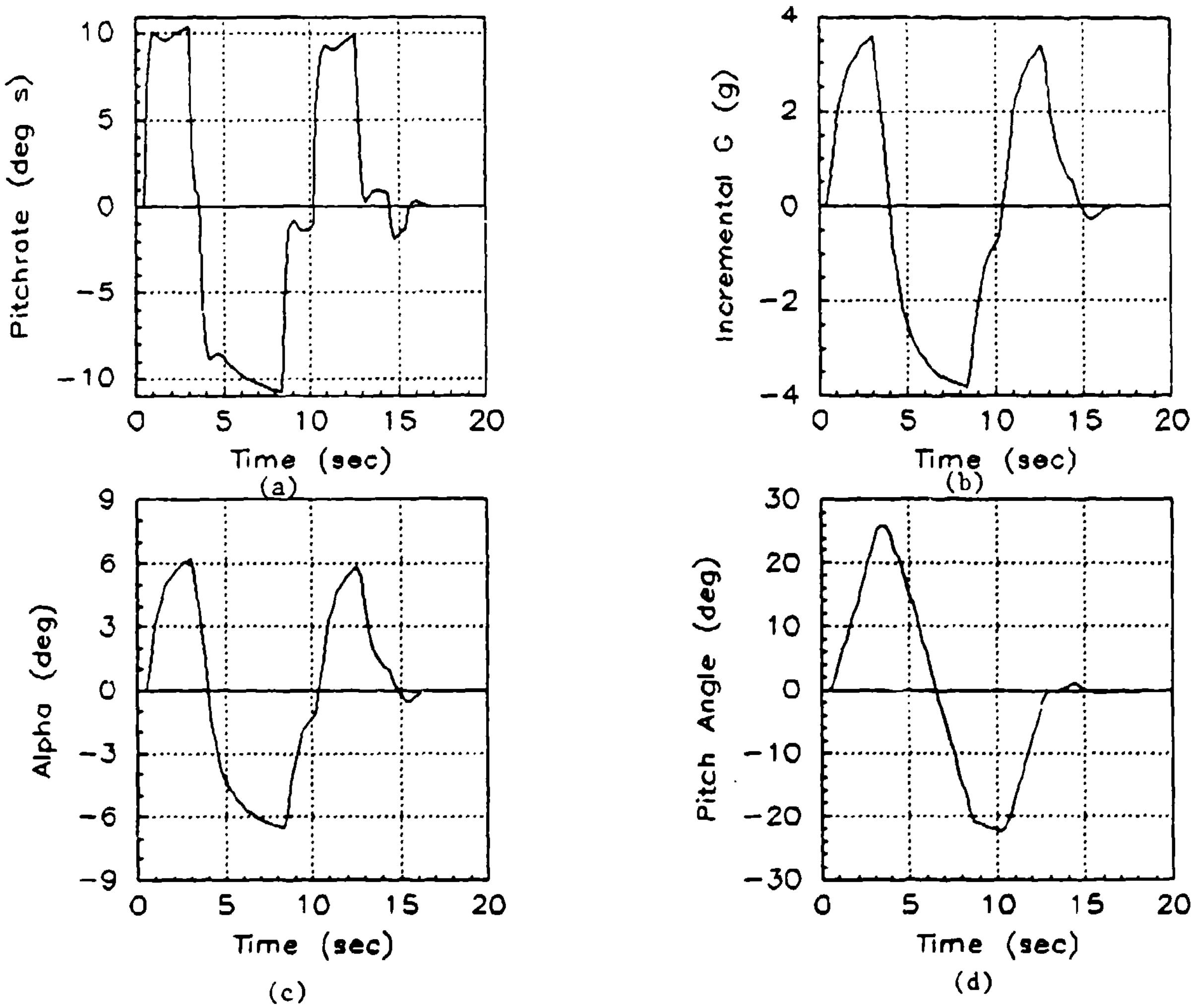


Figure 6.3: Aircraft State Responses for Terrain Avoidance With 'Bang-Bang' Inputs: (a) Pitch Rate, (b) Incremental Load Factor, (c) Angle of Attack, (d) Pitch Angle

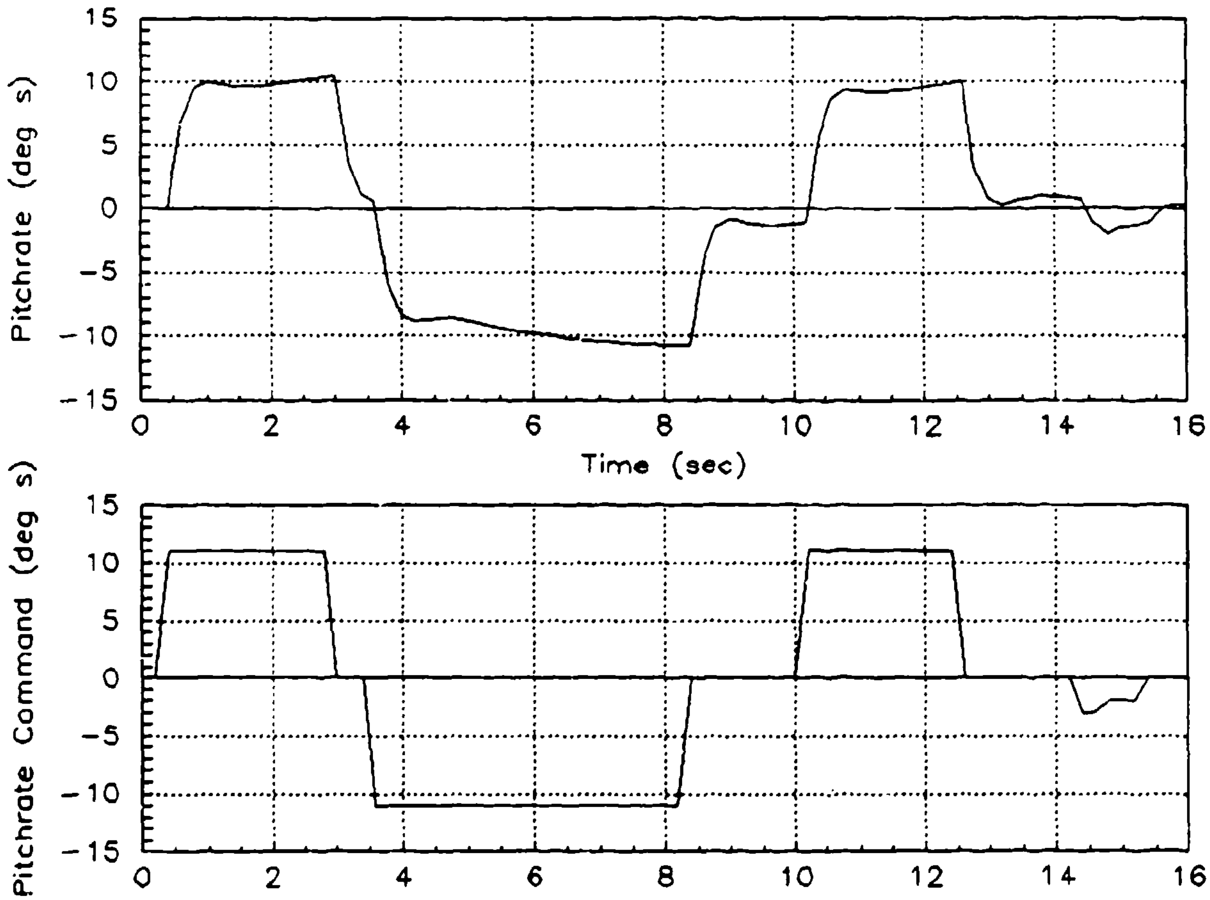


Figure 6.4: Aircraft Pitch Rate Response vs Pitch Rate Command for 'Bang-Bang' Inputs

Appendix A: F-16 Control Derivatives and Trim Conditions

This appendix contains the aerodynamic data used to created the state-space system for the F-16. The data shown on the following page were obtained from test flights of the Advanced Fighter Technology Integration (AFTI)/F-16 technology testbed demonstrator aircraft. While it does have several features that are different from the production F-16, the AFTI aerodynamic data is representative of the normal F-16. It should be noted that the AFTI has a pair of canards mounted on the engine inlet that are 15 degrees off vertical, however, their effect on the aerodynamics of the aircraft is negligible.

Pages A-3 and A-4 of this appendix contain the values of the various longitudinal and lateral-directional control derivatives in both the stability and aircraft body axes. Values for body axis derivatives are given in both dimensional and primed dimensional format. For a discussion of the differences between the two forms, reference Appendix D. Primed dimensional derivatives were used in the construction of the state-space system.

FLIGHT CONDITION

MACH = 0.60 ALTITUDE = 30.00 LOAD FACTOR = 1.00

*** MASS PROPERTIES ***

WEIGHT =	21019.000	IX =	10033.429	IY =	53876.269	IZ =	61278.452	IXZ =	282.13217
MAC =	11.320000	AREA =	300.00000	SPAN =	30.000000	X CG =	35.000000	Z CG =	92.190000
X-ACCEL =	13.950000	VCFLGO =	0.0000000	LGEARO =	0	NEW DATA		ILOADO =	0

*** TRIM SOLUTION ***

ALPHA =	1.5467273	ELEV =	-2.1991414	FLAP =	0.68455285		
CL =	0.13101065	CM =	0.25236194E-02	CD =	0.21780639E-01	QBARO =	532.56117

*** LONGITUDINAL AERO DATA (STAB. AXIS) ***

	CL	CM	CD
ALPHA (DEG)	0.070799	0.004898	0.001524
ELEVATOR (DEG)	0.003993	-0.009920	-0.000258
T.E. FLAP (DEG)	0.015162	-0.000955	0.000645
L.E. FLAP (DEG)	-0.001657	-0.001122	0.000511
Q (RAD)	2.337312	-2.595807	
ALPHA DOT (RAD)	-0.940273	-0.789587	
SNOWPLOW (DEG)	-0.000312	0.000000	0.000000
VELOCITY (FT/S)	-0.000013	-0.000042	0.000008

*** MISC. PARAMETERS ***

NZ/ALPHA = 32.775471

LEF (DEG) = 0.604553

PLOW (DEG) = 0.000000

*** LATERAL-DIRECTIONAL AERO DATA (STAB. AXIS) ***

	CN	CL	CY
BETA (DEG)	0.001761	-0.001600	-0.020953
P (RAD)	-0.004666	-0.230739	0.051413
R (RAD)	-0.473181	-0.004990	0.537411
RUDDER (DEG)	-0.001342	0.000386	0.002920
FLAPERON (DEG)	-0.000147	-0.001954	-0.000094
DIFF TAIL (DEG)	-0.000862	-0.001656	0.001280
CANARD (DEG)	0.001169	0.000144	0.001462

TRIM BETA = 0.000000

TRIM RUDDER = 0.000000

TRIM CANARD = 0.000000

```
*******************************************************
     LONGITUDINAL STABILITY AXIS COEFFICIENTS
              ALPHA =  1.54670
   CL =    .131000            CM =    .000000          CD =    .217810E-01
  CLA =    .707990E-01       CMA =    .489800E-02     CDA =    .152400E-02
 CLDE =    .899300E-02      CMDE =  -.992000E-02     CDDE =  -.258000E-03
 CLDF =    .151620E-01      CMDF =  -.955000E-03     CDDF =    .645000E-03
  CLQ =   2.33731            CMQ =  -2.59581
 CLAD =  -.948273           CMAD =  -.789587
  CLU =  -.130000E-04        CMU =  -.420000E-04      CDU =    .800000E-05
*******************************************************
IS THE ENTERED DATA CORRECT ?   (YES/NO)
YES
*******************************************************
     LONGITUDINAL BODY AXIS COEFFICIENTS  (1/RAD)
   CZ =  -.131540                                      CX =  -.182371E-01
  CZA =  -4.08122            CMA =    .280531         CXA =    .152704
 CZDE =  -.514674                                    CXDE =    .286847E-01
 CZDF =  -.869400                                    CXDF =  -.134941E-01
  CZQ =  -2.33646                                     CXQ =    .630881E-01
 CZAD =    .947582          CMAD =  -.789299         CXAD =  -.255862E-01
  CZU =  -.152964            CMU =  -.761680E-02      CXU =  -.406192E-01
*******************************************************
    LONGITUDINAL AXIS DIMENSIONAL DERIVATIVES
    Z =  -21019.9              M =    .000000           X =  -2914.27
   ZA =  -999.140             MA =   9.41895           XA =   37.3841
  ZDE =  -126.000            MDE =  -19.0834          XDE =   7.02243
  ZDF =  -212.841            MDF =  -1.83716          XDF =  -3.30355
   ZQ =  -4.83500             MQ =  -.736707           XQ =    .130552
  ZAD =   1.96090            MAD =  -.224008          XAD =  -.529473E-01
   ZU =  -.559257E-01         MU =  -.381925E-03       XU =  -.148509E-01
*******************************************************
    LONG BODY AXIS PRIMED DIMENSONAL DERIVATIVES
  ZA' =  -1.49214            MA' =   9.75321          XA' =   37.3841
 ZDE' =  -.188171           MDE' =  -19.0412         XDE' =   7.02243
 ZDF' =  -.317864           MDF' =  -1.76596         XDF' =  -3.30355
  ZQ' =    .992779           MQ' =  -.959097          XQ' =  -17.9453
  ZU' =  -.835210E-04        MU' =  -.363216E-03      XU' =  -.148509E-01
ZTHETA' = -.129799E-02   MTHETA' =   .290760E-03   XTHETA' = -32.1883
*******************************************************
```

```
*****************************************************
      LAT-DIR STABILITY AXIS COEFFICIENTS
  CNB =   .175100E-02      CLB =  -.160800E-02    CYB =  -.209530E-01
  CNP =  -.466600E-02      CLP =  -.330739        CYP =   .514130E-01
  CNR =  -.473181          CLR =  -.499000E-02    CYR =   .537411
 CNDR =  -.134200E-02     CLDR =   .386000E-03   CYDR =   .392000E-02
 CNDA =  -.147000E-03     CLDA =  -.195400E-02   CYDA =  -.940000E-04
CNDDT =  -.862000E-03    CLDDT =  -.165600E-02  CYDDT =   .128000E-02
 CNDC =   .116900E-02     CLDC =   .144000E-03   CYDC =   .146200E-02
*****************************************************
IS THE ENTERED DATA CORRECT ?   (YES/NO)
YES
*****************************************************
       LAT-DIR BODY AXIS COEFFICIENTS
  CNB =   .983743E-01      CLB =  -.948215E-01    CYB =  -1.20052
  CNP =   .188258E-02      CLP =  -.230655        CYP =   .368886E-01
  CNR =  -.473265          CLR =   .155858E-02    CYR =   .538603
 CNDR =  -.762660E-01     CLDR =   .241835E-01   CYDR =   .167304
 CNDA =  -.114413E-01     CLDA =  -.111688       CYDA =  -.538580E-02
CNDDT =  -.519320E-01    CLDDT =  -.935142E-01  CYDDT =   .733386E-01
 CNDC =   .671771E-01     CLDC =   .643971E-02   CYDC =   .837664E-01
*****************************************************
    LAT-DIR BODY AXIS DIMENSIONAL DERIVATIVES
   NB =  7.69604            LB =  -45.3055        YB =  -293.904
   NP =   .329925E-02       LP =  -2.46878        YP =   .202304
   NR =  -.829404           LR =   .166820E-01    YR =   3.95380
  NDR =  -5.96646          LDR =   11.5548       YDR =   40.9583
  NDA =  -.895078          LDA =  -53.3642       YDA =  -1.31852
 NDDT =  -4.06276         LDDT =  -44.6809      YDDT =   17.9543
  NDC =   5.25541          LDC =   3.07688       YDC =   20.5072
*****************************************************
  LAT-DIR BODY AXIS PRIMED DIMENSIONAL DERIVATIVES
  NB' =  7.48842           LB' =  -45.0949       YB' =  -.438925
  NP' =  -.806832E-02      LP' =  -2.46901       YP' =   .272971E-01
  NR' =  -.829434          LR' =  -.664099E-02   YR' =  -.995589
 NDR' =  -5.91402         LDR' =   11.3885      YDR' =   .611684E-01
 NDA' =  -1.14092         LDA' =  -53.3963      YDA' =  -.196912E-02
NDDT' =  -4.26902        LDDT' =  -44.8009     YDDT' =   .268135E-01
 NDC' =   5.27026         LDC' =   3.22508      YDC' =   .306261E-01
*****************************************************
```

Appendix B: F-16 Layout, Sign Conventions, and Axis Definitions

Figure B.1 shows a diagram of the general three-view layout of the F-16. Also contained in this appendix are the definitions of the aircraft axis systems, seen in Figure B.2, and the angles used to differentiate between them. Control surface deflection sign conventions are also shown in Figure B.2 since definitions for positive deflection are not universal.

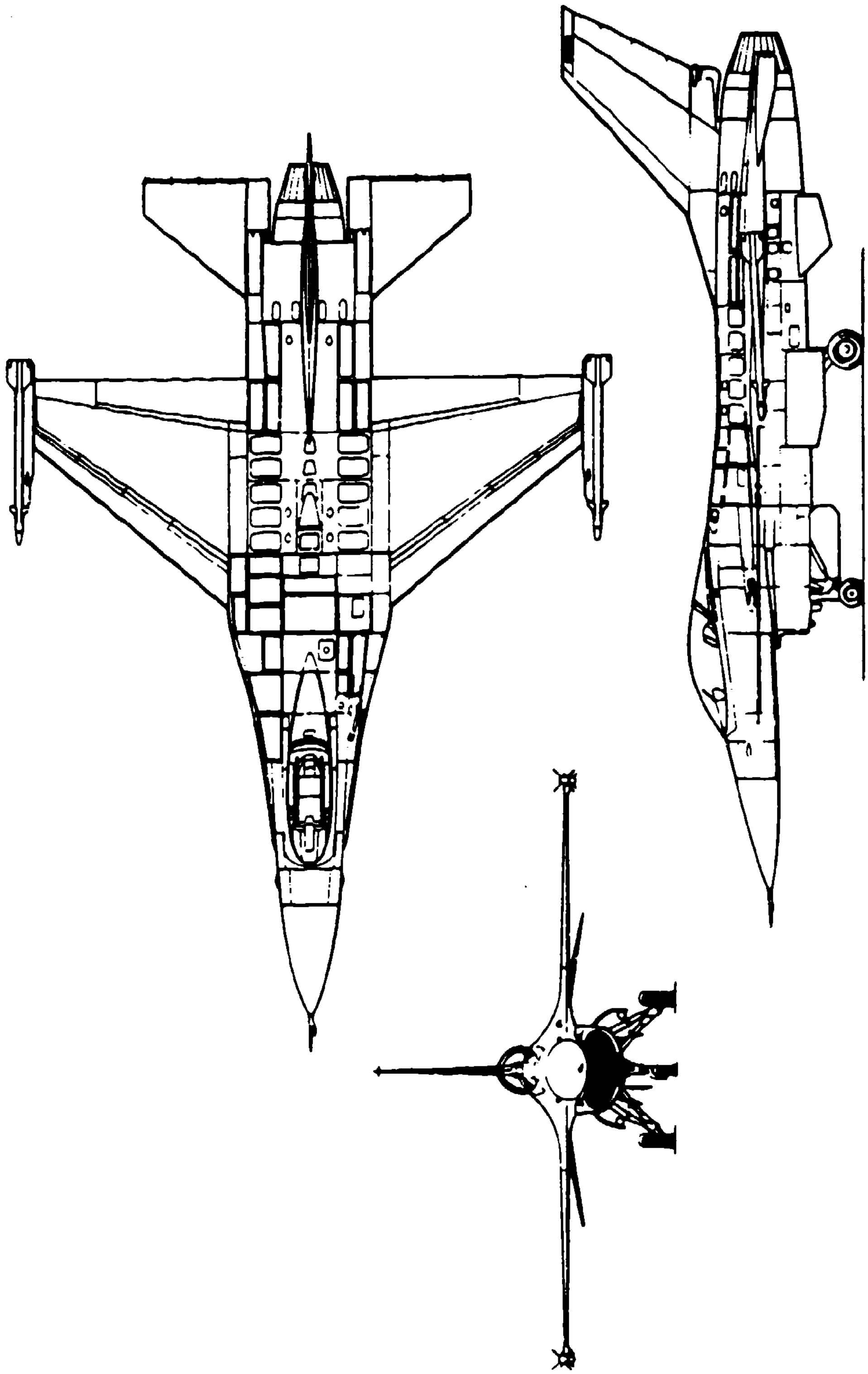
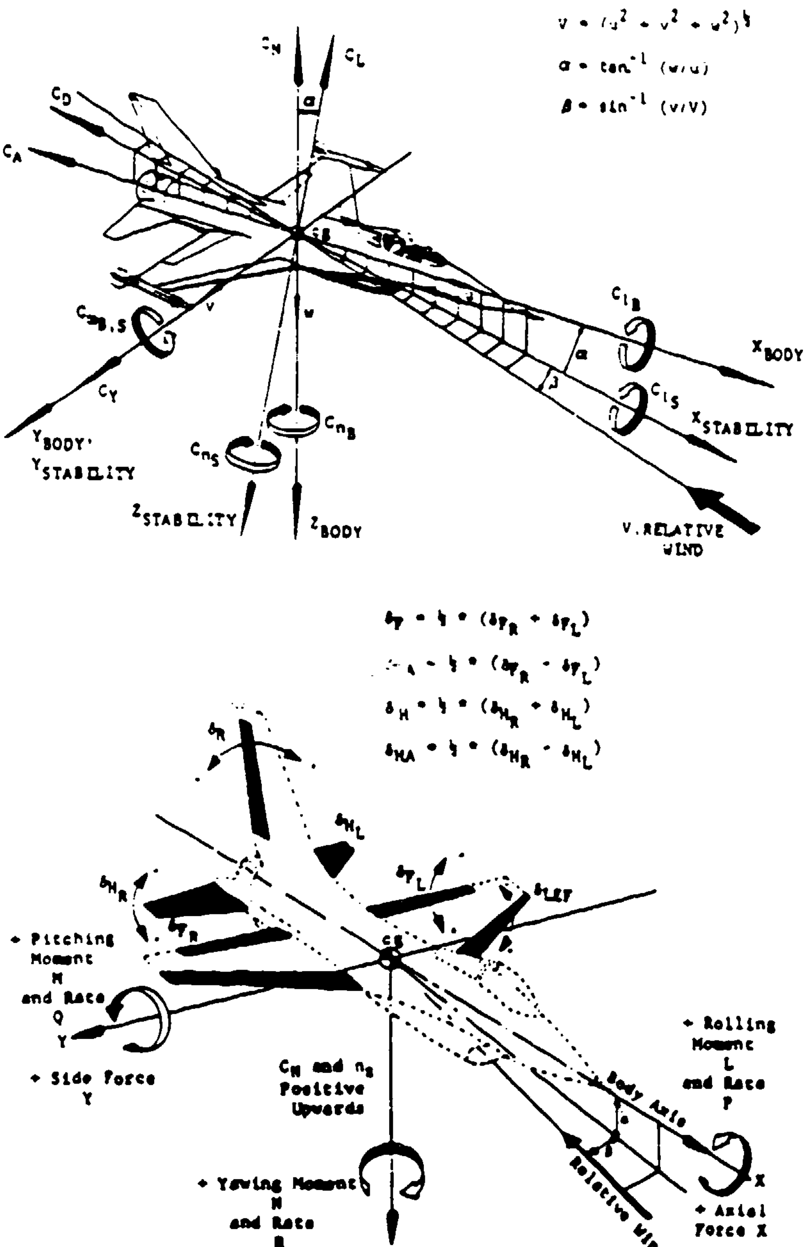


Figure B.1: F-16 Layout and General Arrangement



Notes:

(1) A positive control force produces a negative surface deflection and causes a positive moment about each axis.

(2) For each individual control surface, trailing-edge down (left) is positive.

(3) Leading-edge flap position is measured streamwise. All other positions are measured with respect to the hinge-line.

Figure B.2: F-16 Axis Systems and Sign Conventions

Appendix C: Control Derivative Conversion Program

This appendix contains the computer program used to convert the data listed in Appendix A on page A-2 from the stability axis to the body axis. The program was developed and used in Reference 8. Primed and unprimed dimensional derivatives are also calculated in this program. In addition to values for control derivatives, other parameters such as aircraft mass moments of inertia, trim conditions, and flight conditions are also input. Outputs are shown in Appendix A on pages A-3 and A-4.

```
      PROGRAM CAT
      REAL ALPHA,CL,CLA,CLDE,CLDF,CLQ,CLAD,CLU,
     1CD,CDA,CDDE,CDDF,CDU,
     2CZ,CZA,CZDE,CZDF,CZQ,CZAD,CZU,
     3CX,CXA,CXDE,CXDF,CXU,DALPHA,DPR,
     4CNB,CNP,CNR,
     5CNDR,CNDA,CNDDT,CNDC,
     6CLB,CLP,CLR,
     7CLDR,CLDA,CLDDT,CLDC,
     8CYP,CYR,L,N,
     9M,M1,MA,MAD,MQ,MU,MDE,MDF
      CHARACTER*3 KEY, KEY1, DATA1, DATA2, DATA3, RUN
      DPR = 57.2957795
      WRITE(*,5)
5     FORMAT(1X,'*****************************************************')
      WRITE(*,10)
10    FORMAT(1X,'******** AXIS TRANSFORMATION PROGRAM ***************')
      WRITE(*,20)
20    FORMAT(1X,'*****************************************************')
      WRITE(*,100)
100   FORMAT(1X,'ENTER STABILITY AXIS COEFFICIENTS FOR TRANSFORMATION')
      WRITE(*,101)
101   FORMAT(1X,'TO BODY AXIS.  TRIM ALPHA IS NEEDED FOR CONVERSION.')
      WRITE(*,102)
102   FORMAT(1X,'MOMENT COEFFICIENTS AND SIDEFORCE COEFFICIENTS NOT')
      WRITE(*,40)
40    FORMAT(1X,'REQUESTED REMAIN UNCHANGED.')
      WRITE(*,41)
41    FORMAT(1X,'NOTE:  ALL COEFFICIENTS ARE REQUESTED WHEN COMPUTING')
      WRITE(*,42)
42    FORMAT(1X,'DIMENSIONAL DERIVATIVES.')
103   CONTINUE
      WRITE(*,30)
30    FORMAT(1X,'*****************************************************')
      WRITE(*,106)
106   FORMAT(1X,'TO TRANSFORM ONLY LONGITUDINAL DATA - TYPE LONG')
      WRITE(*,107)
107   FORMAT(1X,'TO TRANSFORM ONLY LATERAL-DIRECTIONAL DATA - TYPE LAT')
      WRITE(*,108)
108   FORMAT(1X,'TO TRANSFORM BOTH LONG AND LAT-DIR DATA - TYPE BOTH')
      WRITE(*,111)
111   FORMAT(1X,'KEYWORD =    ')
      READ(*,109) KEY
109   FORMAT(A3)
      IF(KEY .EQ. 'LAT') GO TO 104
      IF(KEY .EQ. 'LON') GO TO 104
      IF(KEY .EQ. 'BOT') GO TO 104
      GO TO 103
104   CONTINUE
2000  CONTINUE
      WRITE(*,2010)
2010  FORMAT(1X,'ARE DIMENSIONAL BODY AXIS DERIVATIVES REQUIRED ?  (YES/
     1NO)')
      READ(*,2020) KEY1
2020  FORMAT(A3)
      WRITE(*,2030)
2030  FORMAT(1X,'*****************************************************')
      IF (KEY1 .EQ. 'YES') GO TO 2040
      IF (KEY1 .EQ. 'NO ') GO TO 2150
      GO TO 2000
```

```
2040 CONTINUE
     WRITE(*,2050)
2050 FORMAT(1X,'Q  (DYNAMIC PRESSURE - LBS/FT**2) = ')
     READ(*,*) Q
     WRITE(*,2060)
2060 FORMAT(1X,'S  (WING REFERENCE AREA - FT**2) = ')
     READ(*,*) S
     WRITE(*,2065)
2065 FORMAT(1X,'C  (WING MEAN AERODYNAMIC CORD - FT) = ')
     READ(*,*) C
     WRITE(*,2070)
2070 FORMAT(1X,'B  (WING SPAN - FT) = ')
     READ(*,*) B
     WRITE(*,2080)
2080 FORMAT(1X,'VT (TRIM VELOCITY - FT/SEC) = ')
     READ (*,*) U
     WRITE(*,2081)
2081 FORMAT(1X,'THETA (PITCH ANGLE - DEGS) = ')
     READ(*,*) DTHETA
     WRITE(*,2085)
2085 FORMAT(1X,'W  (WEIGHT - LBS) = ')
     READ(*,*) W
     WRITE(*,2090)
2090 FORMAT(1X,'INERTIAS MUST BE INPUT IN BODY AXIS.')
     WRITE(*,2100)
2100 FORMAT(1X,'IXX  (SLUG-FT**2) = ')
     READ(*,*) BIXX
     WRITE(*,2110)
2110 FORMAT(1X,'IYY  (SLUG-FT**2) = ')
     READ(*,*) BIYY
     WRITE(*,2120)
2120 FORMAT(1X,'IZZ  (SLUG-FT**2) = ')
     READ(*,*) BIZZ
     WRITE(*,2130)
2130 FORMAT(1X,'IXZ  (SLUG-FT**2) = ')
     READ(*,*) BIXZ
     WRITE(*,2140)
2140 FORMAT(1X,'*******************************************************')
     WRITE(*,3030)
3030 FORMAT(16X,'AIRCRAFT PARAMETERS')
     WRITE(*,3050) Q
3050 FORMAT(1X,'Q  (DYNAMIC PRESSURE - LBS/FT**2) = ',G13.6)
     WRITE(*,3060) S
3060 FORMAT(1X,'S  (WING REFERENCE AREA - FT**2) = ',G13.6)
     WRITE(*,3065) C
3065 FORMAT(1X,'C  (WING MEAN AERODYNAMIC CORD - FT) = ',G13.6)
     WRITE(*,3070) B
3070 FORMAT(1X,'B  (WING SPAN - FT) = ',G13.6)
     WRITE(*,3080) U
3080 FORMAT(1X,'VT (TRIM VELOCITY - FT/SEC) = ',G13.6)
     WRITE(*,3081) DTHETA
3081 FORMAT(1X,'THETA = ',G13.6)
     WRITE(*,3085) W
3085 FORMAT(1X,'W  (WEIGHT - LBS) = ',G13.6)
     WRITE(*,3100) BIXX
3100 FORMAT(1X,'IXX  (SLUG-FT**2) = ',G13.6)
     WRITE(*,3110) BIYY
3110 FORMAT(1X,'IYY  (SLUG-FT**2) = ',G13.6)
     WRITE(*,3120) BIZZ
3120 FORMAT(1X,'IZZ  (SLUG-FT**2) = ',G13.6)
```

```
      WRITE(*,3130) BIXZ
3130  FORMAT(1X,'IXZ  (SLUG-FT**2) = ',G13.6)
      WRITE(*,3140)
3140  FORMAT(1X,'*****************************************************')
3000  CONTINUE
      WRITE(*,3010)
3010  FORMAT(1X,'IS THE ENTERED DATA CORRECT ?  (YES/NO) ')
      READ(*,3020) DATA3
3020  FORMAT(A3)
      WRITE(*,3025)
3025  FORMAT(1X,'*****************************************************')
      IF(DATA3 .EQ. 'NO ') GO TO 2040
      IF(DATA3 .EQ. 'YES') GO TO 2150
      GO TO 3000
2150  CONTINUE
      WRITE(*,105)
105   FORMAT(1X,'ALPHA (DEG) = ')
      READ(*,*) DALPHA
      ALPHA = DALPHA/DPR
      IF(KEY .EQ. 'LAT')GO TO 460
      WRITE(*,110)
110   FORMAT (1X,'CL = ')
      READ(*,*) CL
      WRITE(*,120)
120   FORMAT(1X,'CLA (1/DEG) = ')
      READ(*,*) CLA
      WRITE(*,130)
130   FORMAT(1X,'CLDE (1/DEG) = ')
      READ(*,*) CLDE
      WRITE(*,140)
140   FORMAT(1X,'CLDF (1/DEG) = ')
      READ(*,*) CLDF
      WRITE(*,150)
150   FORMAT(1X,'CLQ (1/RAD) = ')
      READ(*,*) CLQ
      WRITE(*,160)
160   FORMAT(1X,'CLAD (1/RAD) = ')
      READ(*,*) CLAD
      WRITE(*,170)
170   FORMAT(1X,'CLU (1/(FT/SEC)) = ')
      READ(*,*) CLU
      WRITE(*,180)
180   FORMAT(1X,'CD = ')
      READ(*,*) CD
      WRITE(*,190)
190   FORMAT(1X,'CDA (1/DEG) = ')
      READ(*,*) CDA
      WRITE(*,200)
200   FORMAT(1X,'CDDE (1/DEG) = ')
      READ(*,*) CDDE
      WRITE(*,210)
210   FORMAT(1X,'CDDF (1/DEG) = ')
      READ(*,*) CDDF
      WRITE(*,220)
220   FORMAT(1X,'CDU (1/(FT/SEC)) = ')
      READ(*,*) CDU
      WRITE(*,1000)
1000  FORMAT(1X,'CM = ')
      READ(*,*) CM
      WRITE(*,1010)
```

```
1010 FORMAT(1X,'CMA (1/DEG) = ')
     READ(*,*) CMA
     IF (KEY1 .EQ. 'NO ') GO TO 1005
     WRITE(*,1030)
1030 FORMAT(1X,'CMDE (1/DEG) = ')
     READ(*,*) CMDE
     WRITE(*,1040)
1040 FORMAT(1X,'CMDF (1/DEG) = ')
     READ(*,*) CMDF
     WRITE(*,1050)
1050 FORMAT(1X,'CMQ (1/RAD) = ')
     READ(*,*) CMQ
1005 CONTINUE
     WRITE(*,1060)
1060 FORMAT(1X,'CMAD (1/RAD) = ')
     READ(*,*) CMAD
     WRITE(*,1020)
1020 FORMAT(1X,'CMU (1/(FT/SEC)) = ')
     READ(*,*) CMU
     WRITE(*,225)
225  FORMAT(1X,'*****************************************************')
     WRITE(*,226)
226  FORMAT(6X,'LONGITUDINAL STABILITY AXIS COEFFICIENTS')
     WRITE(*,230) DALPHA
230  FORMAT(15X,'ALPHA =',G13.6)
     IF (KEY1 .EQ. 'YES') GO TO 1080
     WRITE(*,240) CL,CD
240  FORMAT(1X,'CL =     ',G13.6,6X,'CD = ',G13.6)
     WRITE(*,250) CLA,CDA
250  FORMAT(1X,'CLA =    ',G13.6,5X,'CDA = ',G13.6)
     WRITE(*,260) CLDE,CDDE
260  FORMAT(1X,'CLDE =   ',G13.6,4X,'CDDE = ',G13.6)
     WRITE(*,270) CLDF,CDDF
270  FORMAT(1X,'CLDF =   ',G13.6,4X,'CDDF = ',G13.6)
     WRITE(*,280) CLQ
280  FORMAT(1X,'CLQ =    ',G13.6)
     WRITE(*,290) CLAD
290  FORMAT(1X,'CLAD =   ',G13.6)
     WRITE(*,300) CLU,CDU
300  FORMAT(1X,'CLU =    ',G13.6,5X,'CDU = ',G13.6)
1080 CONTINUE
     IF(KEY1 .EQ. 'NO ') GO TO 1170
     WRITE(*,1090) CL,CM,CD
1090 FORMAT(4X,'CL = ',G13.6,9X,'CM = ',G13.6,6X,'CD = ',G13.6)
     WRITE(*,1100) CLA,CMA,CDA
1100 FORMAT(3X,'CLA = ',G13.6,8X,'CMA = ',G13.6,5X,'CDA = ',G13.6)
     WRITE(*,1110) CLDE,CMDE,CDDE
1110 FORMAT(2X,'CLDE = ',G13.6,7X,'CMDE = ',G13.6,4X,'CDDE = ',G13.6)
     WRITE(*,1120) CLDF,CMDF,CDDF
1120 FORMAT(2X,'CLDF = ',G13.6,7X,'CMDF = ',G13.6,4X,'CDDF = ',G13.6)
     WRITE(*,1130) CLQ,CMQ
1130 FORMAT(3X,'CLQ = ',G13.6,8X,'CMQ = ',G13.6)
     WRITE(*,1150) CLAD,CMAD
1150 FORMAT(2X,'CLAD = ',G13.6,7X,'CMAD = ',G13.6)
     WRITE(*,1140) CLU,CMU,CDU
1140 FORMAT(3X,'CLU = ',G13.6,8X,'CMU = ',G13.6,5X,'CDU = ',G13.6)
1170 CONTINUE
     WRITE(*,310)
310  FORMAT(1X,'*****************************************************')
315  CONTINUE
```

```
      WRITE(*,320)
 320  FORMAT(1X,'IS THE ENTERED DATA CORRECT ?   (YES/NO)')
      READ(*,330) DATA1
 330  FORMAT(A3)
      WRITE(*,335)
 335  FORMAT(1X,'*************************************************************')
      IF(DATA1 .EQ. 'NO ') GO TO 2150
      IF(DATA1 .EQ. 'YES') GO TO 340
      GO TO 315
 340  CONTINUE
      WRITE(*,345)
 345  FORMAT(6X,'LONGITUDINAL BODY AXIS COEFFICIENTS  (1/RAD)')
C
      CLA = CLA*DPR
      CLDE = CLDE*DPR
      CLDF = CLDF*DPR
      CDA = CDA*DPR
      CDDE = CDDE*DPR
      CDDF = CDDF*DPR
      CMA = CMA*DPR
C
      IF (KEY1 .EQ. 'NO ') GO TO 346
C
      CMDE = CMDE*DPR
      CMDF = CMDF*DPR
C
 346  CONTINUE
C
      SCZA = -CLA - CD
      SCZAD = -CLAD
      SCZQ = -CLQ
      SCZU = -CLU - 2.0*CL
      SCZDE = -CLDE
      SCZDF = -CLDF
      SCXA = -CDA + CL
      SCXU = -CDU - 2.0*CD
      SCXDE = -CDDE
      SCXDF = -CDDF
C
      CAL = COS(ALPHA)
      SAL = SIN(ALPHA)
      COSSQ = CAL**2
      SINSQ = SAL**2
      COSSIN = CAL*SAL
C
      CZ = -CL*CAL - CD*SAL
      CZA = SCZA*COSSQ +(SCZU+SCXA)*COSSIN + SCXU*SINSQ
      CZAD = SCZAD*COSSQ
      CZQ = SCZQ*CAL
      CZU = SCZU*COSSQ - (SCZA-SCXU)*COSSIN - SCXA*SINSQ
      CZDE = SCZDE*CAL + SCXDE*SAL
      CZDF = SCZDF*CAL + SCXDF*SAL
C
      CX = -CD*CAL + CL*SAL
      CXA = SCXA*COSSQ + (SCXU-SCZA)*COSSIN - SCZU*SINSQ
      CXAD = CLAD*COSSIN
      CXQ = CLQ*SAL
      CXU = SCXU*COSSQ - (SCXA+SCZU)*COSSIN + SCZA*SINSQ
      CXDE = SCXDE*CAL - SCZDE*SAL
      CXDF = SCXDF*CAL - SCZDF*SAL
```

```
C
      BCMA = CMA*CAL + (CMU + 2.0*CM)*SAL
      BCMAD = CMAD*CAL
      BCMU = (CMU + 2.0*CM)*CAL - CMA*SAL
C
      WRITE(*,350) CZ,CX
 350  FORMAT(4X,'CZ = ',G13.6,33X,'CX = ',G13.6)
      WRITE(*,360) CZA,BCMA,CXA
 360  FORMAT(3X,'CZA = ',G13.6,8X,'CMA = ',G13.6,5X,'CXA = ',G13.6)
      WRITE(*,370) CZDE,CXDE
 370  FORMAT(2X,'CZDE = ',G13.6,31X,'CXDE = ',G13.6)
      WRITE(*,380) CZDF,CXDF
 380  FORMAT(2X,'CZDF = ',G13.6,31X,'CXDF = ',G13.6)
      WRITE(*,390) CZQ,CXQ
 390  FORMAT(3X,'CZQ = ',G13.6,32X,'CXQ = ',G13.6)
      WRITE(*,400) CZAD,BCMAD,CXAD
 400  FORMAT(2X,'CZAD = ',G13.6,7X,'CMAD = ',G13.6,4X,'CXAD = ',G13.6)
      WRITE(*,410) CZU,BCMU,CXU
 410  FORMAT(3X,'CZU = ',G13.6,8X,'CMU = ',G13.6,5X,'CXU = ',G13.6)
      WRITE(*,420)
 420  FORMAT(1X,'*****************************************************')
      IF (KEY1 .EQ. 'NO ') GO TO 1360
      Z1 = (Q*S*32.2)/W
      A = C/(2.0*U)
      THETA = DTHETA/DPR
C
      Z = Q*S*CZ
      ZA = Z1*CZA
      ZAD = Z1*A*CZAD
      ZQ = Z1*A*CZQ
      ZU = (Z1/U)*CZU
      ZDE = Z1*CZDE
      ZDF = Z1*CZDF
C
      X = Q*S*CX
      XA = Z1*CXA
      XAD = Z1*A*CXAD
      XQ = Z1*A*CXQ
      XU = (Z1/U)*CXU
      XDE = Z1*CXDE
      XDF = Z1*CXDF
C
      M1 = (Q*S*C)/BIYY
C
      M = Q*S*C*CM
      MA = M1*BCMA
      MAD = M1*A*BCMAD
      MQ = M1*A*CMQ
      MU = (M1/U)*BCMU
      MDE = M1*CMDE
      MDF = M1*CMDF
C
      WRITE(*,1180)
 1180 FORMAT (5X,'LONGITUDINAL AXIS DIMENSIONAL DERIVATIVES')
      WRITE(*,1190) Z,M,X
 1190 FORMAT(5X,'Z = ',G13.6,10X,'M = ',G13.6,7X,'X = ',G13.6)
      WRITE(*,1200) ZA,MA,XA
 1200 FORMAT(4X,'ZA = ',G13.6,9X,'MA = ',G13.6,6X,'XA = ',G13.6)
      WRITE(*,1210) ZDE,MDE,XDE
 1210 FORMAT(3X,'ZDE = ',G13.6,8X,'MDE = ',G13.6,5X,'XDE = ',G13.6)
```

```
      WRITE(*,1220) ZDF,MDF,XDF
 1220 FORMAT(3X,'ZDF = ',G13.6,8X,'MDF = ',G13.6,5X,'XDF = ',G13.6)
      WRITE(*,1230) ZQ,MQ,XQ
 1230 FORMAT(4X,'ZQ = ',G13.6,9X,'MQ = ',G13.6,6X,'XQ = ',G13.6)
      WRITE(*,1250) ZAD,MAD,XAD
 1250 FORMAT(3X,'ZAD = ',G13.6,8X,'MAD = ',G13.6,5X,'XAD = ',G13.6)
      WRITE(*,1240) ZU,MU,XU
 1240 FORMAT(4X,'ZU = ',G13.6,9X,'MU = ',G13.6,6X,'XU = ',G13.6)
      WRITE(*,1260)
 1260 FORMAT(1X,'*****************************************************')
C
      PZA = ZA/U
      PZQ = (ZQ/U) + 1.0
      PZU = ZU/U
      PZDE = ZDE/U
      PZDF = ZDF/U
      PZTHETA = -(32.2/U)*SIN(THETA)
C
      PMA = MA + MAD*PZA
      PMQ = MQ + MAD*PZQ
      PMU = MU + MAD*PZU
      PMDE = MDE + MAD*PZDE
      PMDF = MDF + MAD*PZDF
      PMTHETA = MAD*PZTHETA
C
      PXQ = XQ - U*ALPHA
      PXTHETA = -32.2*COS(THETA)
      WRITE(*,1280)
 1280 FORMAT(5X,'LONG BODY AXIS PRIMED DIMENSONAL DERIVATIVES')
      WRITE(*,1290) PZA,PMA,XA
 1290 FORMAT(3X,'ZA'' = ',G13.6,8X,'MA'' = ',G13.6,5X,'XA'' = ',G13.6)
      WRITE(*,1300) PZDE,PMDE,XDE
 1300 FORMAT(2X,'ZDE'' = ',G13.6,7X,'MDE'' = ',G13.6,4X,'XDE'' = ',G13.6
     +)
      WRITE(*,1310) PZDF,PMDF,XDF
 1310 FORMAT(2X,'ZDF'' = ',G13.6,7X,'MDF'' = ',G13.6,4X,'XDF'' = ',G13.6
     +)
      WRITE(*,1320) PZQ,PMQ,PXQ
 1320 FORMAT(3X,'ZQ'' = ',G13.6,8X,'MQ'' = ',G13.6,5X,'XQ'' = ',G13.6)
      WRITE(*,1330) PZU,PMU,XU
 1330 FORMAT(3X,'ZU'' = ',G13.6,8X,'MU'' = ',G13.6,5X,'XU'' = ',G13.6)
      WRITE(*,1340) PZTHETA,PMTHETA,PXTHETA
 1340 FORMAT(1X,'ZTHETA'' = ',G12.6,4X,'MTHETA'' = ',G12.6,3X,'XTHETA''
     += ',G12.6)
      WRITE(*,1350)
 1350 FORMAT(1X,'*****************************************************')
 1360 CONTINUE
      IF(KEY .EQ. 'BOT') GO TO 446
 421  CONTINUE
      WRITE(*,430)
 430  FORMAT(1X,'IS ANOTHER PROGRAM RUN DESIRED ?  (YES/NO)')
      READ(*,440) RUN
 440  FORMAT(A3)
      WRITE(*,445)
 445  FORMAT(1X,'*****************************************************')
      IF(RUN .EQ. 'NO ') GO TO 450
      IF(RUN .EQ. 'YES') GO TO 103
      GO TO 421
 446  CONTINUE
      WRITE(*,447)
```

```
447   FORMAT(1X,'****************************************************************')
460   CONTINUE
      WRITE(*,455)
455   FORMAT(1X,'CNB (1/DEG) = ')
      READ(*,*) CNB
      WRITE(*,470)
470   FORMAT(1X,'CNP (1/RAD) = ')
      READ(*,*) CNP
      WRITE(*,480)
480   FORMAT(1X,'CNR (1/RAD) = ')
      READ(*,*) CNR
      WRITE(*,490)
490   FORMAT(1X,'CNDR (1/DEG) = ')
      READ(*,*) CNDR
      WRITE(*,500)
500   FORMAT(1X,'CNDA (1/DEG) = ')
      READ(*,*) CNDA
      WRITE(*,510)
510   FORMAT(1X,'CNDDT (1/DEG) = ')
      READ(*,*) CNDDT
      WRITE(*,520)
520   FORMAT(1X,'CNDC (1/DEG) = ')
      READ(*,*) CNDC
      WRITE(*,530)
530   FORMAT(1X,'CLB (1/DEG) = ')
      READ(*,*) CLB
      WRITE(*,540)
540   FORMAT(1X,'CLP (1/RAD) = ')
      READ(*,*) CLP
      WRITE(*,550)
550   FORMAT(1X,'CLR (1/RAD) = ')
      READ(*,*) CLR
      WRITE(*,560)
560   FORMAT(1X,'CLDR (1/DEG) = ')
      READ(*,*) CLDR
      WRITE(*,570)
570   FORMAT(1X,'CLDA (1/DEG) = ')
      READ(*,*) CLDA
      WRITE(*,580)
580   FORMAT(1X,'CLDDT (1/DEG) = ')
      READ(*,*) CLDDT
      WRITE(*,590)
590   FORMAT(1X,'CLDC (1/DEG) = ')
      READ(*,*) CLDC
      IF (KEY1 .EQ. 'NO ') GO TO 609
      WRITE(*,611)
611   FORMAT(1X,'CYB (1/DEG) = ')
      READ(*,*) CYB
609   CONTINUE
      WRITE(*,600)
600   FORMAT(1X,'CYP (1/RAD) = ')
      READ(*,*) CYP
      WRITE(*,610)
610   FORMAT(1X,'CYR (1/RAD) = ')
      READ(*,*) CYR
      IF(KEY1 .EQ. 'NO ') GO TO 616
      WRITE(*,612)
612   FORMAT(1X,'CYDR (1/DEG) = ')
      READ(*,*) CYDR
      WRITE(*,613)
```

```
613  FORMAT(1X,'CYDA (1/DEG) = ')
     READ(*,*) CYDA
     WRITE(*,614)
614  FORMAT(1X,'CYDDT (1/DEG) = ')
     READ(*,*) CYDDT
     WRITE(*,615)
615  FORMAT(1X,'CYDC (1/DEG) = ')
     READ(*,*) CYDC
616  CONTINUE
     WRITE(*,620)
620  FORMAT(1X,'*****************************************************')
     WRITE(*,630)
630  FORMAT(8X,'LAT-DIR STABILITY AXIS COEFFICIENTS')
     IF(KEY .EQ. 'LON') GO TO 635
     IF(KEY .EQ. 'BOT') GO TO 635
     WRITE(*,631) DALPHA
631  FORMAT(15X,'ALPHA = ',G13.6)
635  CONTINUE
     IF(KEY1 .EQ. 'YES') GO TO 711
     WRITE(*,640) CNB,CLB
640  FORMAT(3X,'CNB = ',G13.6,8X,'CLB = ',G13.6)
     WRITE(*,650) CNP,CLP,CYP
650  FORMAT(3X,'CNP = ',G13.6,8X,'CLP = ',G13.6,5X,'CYP = ',G13.6)
     WRITE(*,660) CNR,CLR,CYR
660  FORMAT(3X,'CNR = ',G13.6,8X,'CLR = ',G13.6,5X,'CYR = ',G13.6)
     WRITE(*,670) CNDR,CLDR
670  FORMAT(2X,'CNDR = ',G13.6,7X,'CLDR = ',G13.6)
     WRITE(*,680) CNDA,CLDA
680  FORMAT(2X,'CNDA = ',G13.6,7X,'CLDA = ',G13.6)
     WRITE(*,690) CNDDT,CLDDT
690  FORMAT(1X,'CNDDT = ',G13.6,6X,'CLDDT = ',G13.6)
     WRITE(*,700) CNDC,CLDC
700  FORMAT(2X,'CNDC = ',G13.6,7X,'CLDC = ',G13.6)
     WRITE(*,710)
710  FORMAT(1X,'*****************************************************')
     IF(KEY1 .EQ. 'NO ') GO TO 720
711  CONTINUE
     WRITE(*,712) CNB,CLB,CYB
712  FORMAT(3X,'CNB = ',G13.6,8X,'CLB = ',G13.6,5X,'CYB = ',G13.6)
     WRITE(*,713) CNP,CLP,CYP
713  FORMAT(3X,'CNP = ',G13.6,8X,'CLP = ',G13.6,5X,'CYP = ',G13.6)
     WRITE(*,714) CNR,CLR,CYR
714  FORMAT(3X,'CNR = ',G13.6,8X,'CLR = ',G13.6,5X,'CYR = ',G13.6)
     WRITE(*,715) CNDR,CLDR,CYDR
715  FORMAT(2X,'CNDR = ',G13.6,7X,'CLDR = ',G13.6,4X,'CYDR = ',G13.6)
     WRITE(*,716) CNDA,CLDA,CYDA
716  FORMAT(2X,'CNDA = ',G13.6,7X,'CLDA = ',G13.6,4X,'CYDA = ',G13.6)
     WRITE(*,717) CNDDT,CLDDT,CYDDT
717  FORMAT(1X,'CNDDT = ',G13.6,6X,'CLDDT = ',G13.6,3X,'CYDDT = ',G13.6
    +)
     WRITE(*,718) CNDC,CLDC,CYDC
718  FORMAT(2X,'CNDC = ',G13.6,7X,'CLDC = ',G13.6,4X,'CYDC = ',G13.6)
     WRITE(*,719)
719  FORMAT(1X,'*****************************************************')
720  CONTINUE
     WRITE(*,730)
730  FORMAT(1X,'IS THE ENTERED DATA CORRECT ?   (YES/NO)')
     READ(*,740) DATA2
740  FORMAT(A3)
     WRITE(*,750)
```

```
750  FORMAT(1X,'****************************************************')
     IF(KEY .EQ. 'BOT') GO TO 755
     IF(DATA2 .EQ. 'NO ') GO TO 2150
     IF(DATA2 .EQ. 'YES') GO TO 760
     GO TO 720
755  CONTINUE
     IF(DATA2 .EQ. 'NO ') GO TO 460
     IF(DATA2 .EQ. 'YES') GO TO 760
     GO TO 720
760  CONTINUE
     CNB=CNB*DPR
     CNDR=CNDR*DPR
     CNDA =CNDA*DPR
     CNDDT=CNDDT*DPR
     CNDC=CNDC*DPR
     CLB=CLB*DPR
     CLDR=CLDR*DPR
     CLDA=CLDA*DPR
     CLDDT=CLDDT*DPR
     CLDC=CLDC*DPR
     IF(KEY1 .EQ. 'NO ') GO TO 765
     CYB = CYB*DPR
     CYDR = CYDR*DPR
     CYDA = CYDA*DPR
     CYDDT = CYDDT*DPR
     CYDC = CYDC*DPR
 765 CONTINUE
     BCLB = CLB*COS(ALPHA)-CNB*SIN(ALPHA)
     BCLP=CLP*COS(ALPHA)**2-(CLR+CNP)*SIN(ALPHA)*COS(ALPHA)+CNR*SIN(ALP
    1HA)**2
     BCLR=CLR*COS(ALPHA)**2-(CNR-CLP)*SIN(ALPHA)*COS(ALPHA)-CNP*SIN(ALP
    1HA)**2
     BCLDA = CLDA*COS(ALPHA)-CNDA*SIN(ALPHA)
     BCLDR = CLDR*COS(ALPHA)-CNDR*SIN(ALPHA)
     BCLDC = CLDC*COS(ALPHA)-CNDC*SIN(ALPHA)
     BCLDDT = CLDDT*COS(ALPHA)-CNDDT*SIN(ALPHA)
     BCNB = CNB*COS(ALPHA)+CLB*SIN(ALPHA)
     BCNP = CNP*COS(ALPHA)**2-(CNR-CLP)*SIN(ALPHA)*COS(ALPHA)-CLR*SIN(A
    1LPHA)**2
     BCNR = CNR*COS(ALPHA)**2+(CLR+CNP)*SIN(ALPHA)*COS(ALPHA)+CLP*SIN(A
    1LPHA)**2
     BCNDA = CNDA*COS(ALPHA)+CLDA*SIN(ALPHA)
     BCNDR = CNDR*COS(ALPHA)+CLDR*SIN(ALPHA)
     BCNDC = CNDC*COS(ALPHA)+CLDC*SIN(ALPHA)
     BCNDDT = CNDDT*COS(ALPHA)+CLDDT*SIN(ALPHA)
     BCYR = CYR*COS(ALPHA)+CYP*SIN(ALPHA)
     BCYP = CYP*COS(ALPHA)-CYR*SIN(ALPHA)
     WRITE(*,770)
770  FORMAT(9X,'LAT-DIR BODY AXIS COEFFICIENTS')
     WRITE(*,780) BCNB,BCLB,CYB
780  FORMAT(3X,'CNB = ',G13.6,8X,'CLB = ',G13.6,5X,'CYB = ',G13.6)
     WRITE(*,790) BCNP,BCLP,BCYP
790  FORMAT(3X,'CNP = ',G13.6,8X,'CLP = ',G13.6,5X,'CYP = ',G13.6)
     WRITE(*,800) BCNR,BCLR,BCYR
800  FORMAT(3X,'CNR = ',G13.6,8X,'CLR = ',G13.6,5X,'CYR = ',G13.6)
     WRITE(*,810) BCNDR,BCLDR,CYDR
810  FORMAT(2X,'CNDR = ',G13.6,7X,'CLDR = ',G13.6,4X,'CYDR = ',G13.6)
     WRITE(*,820) BCNDA,BCLDA,CYDA
820  FORMAT(2X,'CNDA = ',G13.6,7X,'CLDA = ',G13.6,4X,'CYDA = ',G13.6)
     WRITE(*,830) BCNDDT,BCLDDT,CYDDT
```

```
830  FORMAT(1X,'CNDDT = ',G13.6,6X,'CLDDT = ',G13.6,3X,'CYDDT = ',G13.6
    *)
     WRITE(*,840) BCNDC,BCLDC,CYDC
840  FORMAT(2X,'CNDC = ',G13.6,7X,'CLDC = ',G13.6,4X,'CYDC = ',G13.6)
     WRITE(*,850)
850  FORMAT(1X,'*******************************************************')
     IF (KEY1 .EQ. 'NO ') GO TO 421
     N = (Q*S*B)/BIZZ
     L = (Q*S*B)/BIXX
     B = B/(2.0*U)
     Y = (Q*S*32.2)/W
     BNB = N*BCNB
     BNP = N*B*BCNP
     BNR = N*B*BCNR
     BNDR = N*BCNDR
     BNDA = N*BCNDA
     BNDDT = N*BCNDDT
     BNDC = N*BCNDC
     BLB = L*BCLB
     BLP = L*B*BCLP
     BLR = L*B*BCLR
     BLDR = L*BCLDR
     BLDA = L*BCLDA
     BLDDT = L*BCLDDT
     BLDC = L*BCLDC
     YB = Y*CYB
     BYR = Y*B*BCYR
     BYP = Y*B*BCYP
     YDR = Y*CYDR
     YDA = Y*CYDA
     YDDT = Y*CYDDT
     YDC = Y*CYDC
     WRITE(*,2160)
2160 FORMAT(5X,'LAT-DIR BODY AXIS DIMENSIONAL DERIVATIVES')
     WRITE(*,2170) BNB,BLB,YB
2170 FORMAT(4X,'NB = ',G13.6,9X,'LB = ',G13.6,5X,'YB = ',G13.6)
     WRITE(*,2180) BNP,BLP,BYP
2180 FORMAT(4X,'NP = ',G13.6,9X,'LP = ',G13.6,5X,'YP = ',G13.6)
     WRITE(*,2190) BNR,BLR,BYR
2190 FORMAT(4X,'NR = ',G13.6,9X,'LR = ',G13.6,5X,'YR = ',G13.6)
     WRITE(*,2200) BNDR,BLDR,YDR
2200 FORMAT(3X,'NDR = ',G13.6,8X,'LDR = ',G13.6,4X,'YDR = ',G13.6)
     WRITE(*,2210) BNDA,BLDA,YDA
2210 FORMAT(3X,'NDA = ',G13.6,8X,'LDA = ',G13.6,4X,'YDA = 'G13.6)
     WRITE(*,2220) BNDDT,BLDDT,YDDT
2220 FORMAT(2X,'NDDT = ',G13.6,7X,'LDDT = ',G13.6,3X,'YDDT = ',G13.6)
     WRITE(*,2230) BNDC,BLDC,YDC
2230 FORMAT(3X,'NDC = ',G13.6,8X,'LDC = ',G13.6,4X,'YDC = ',G13.6)
     WRITE(*,2240)
2240 FORMAT(1X,'*******************************************************')
     D = 1.0 - ((BIXZ*BIXZ)/(BIXX*BIZZ))
     R1 = BIXZ/BIZZ
     R2 = BIXZ/BIXX
     PBNB = (BNB + R1*BLB)/D
     PBNP = (BNP + R1*BLP)/D
     PBNR = (BNR + R1*BLR)/D
     PBNDR = (BNDR + R1*BLDR)/D
     PBNDA = (BNDA + R1*BLDA)/D
     PBNDDT = (BNDDT + R1*BLDDT)/D
     PBNDC = (BNDC + R1*BLDC)/D
```

```
      PBLB = (BLB + R2*BNB)/D
      PBLP = (BLP + R2*BNP)/D
      PBLR = (BLR + R2*BNR)/D
      PBLDR = (BLDR + R2*BNDR)/D
      PBLDA = (BLDA + R2*BNDA)/D
      PBLDDT = (BLDDT + R2*BNDDT)/D
      PBLDC = (BLDC + R2*BNDC)/D
      PYB = YB/U
      PBYP = BYP/U + ALPHA
      PBYR = BYR/U - 1.0
      PYDR = YDR/U
      PYDA = YDA/U
      PYDDT = YDDT/U
      PYDC = YDC/U
      WRITE(*,2250)
 2250 FORMAT(3X,'LAT-DIR BODY AXIS PRIMED DIMENSIONAL DERIVATIVES')
      WRITE(*,2260) PBNB,PBLB,PYB
 2260 FORMAT(3X,'NB'' = ',G13.6,8X,'LB'' = ',G13.6,4X,'YB'' = ',G13.6)
      WRITE(*,2270) PBNP,PBLP,PBYP
 2270 FORMAT(3X,'NP'' = ',G13.6,8X,'LP'' = ',G13.6,4X,'YP'' = ',G13.6)
      WRITE(*,2280) PBNR,PBLR,PBYR
 2280 FORMAT(3X,'NR'' = ',G13.6,8X,'LR'' = ',G13.6,4X,'YR'' = ',G13.6)
      WRITE(*,2290) PBNDR,PBLDR,PYDR
 2290 FORMAT(2X,'NDR'' = ',G13.6,7X,'LDR'' = ',G13.6,3X,'YDR'' = ',G13.6
     +)
      WRITE(*,2300) PBNDA,PBLDA,PYDA
 2300 FORMAT(2X,'NDA'' = ',G13.6,7X,'LDA'' = ',G13.6,3X,'YDA'' = ',G13.6
     +)
      WRITE(*,2310) PBNDDT,PBLDDT,PYDDT
 2310 FORMAT(1X,'NDDT'' = ',G13.6,6X,'LDDT'' = ',G13.6,2X,'YDDT'' = ',G1
     +3.6)
      WRITE(*,2320) PBNDC,PBLDC,PYDC
 2320 FORMAT(2X,'NDC'' = ',G13.6,7X,'LDC'' = ',G13.6,3X,'YDC'' = ',G13.6
     +)
      WRITE(*,2340)
 2340 FORMAT(1X,'*************************************************************')
      GO TO 421
  450 CONTINUE
      END
```

Appendix D: Linearized Equations of Motion

The linearized equations of motion derived in this appendix are used to create the state-space system for the F-16. It should be duly noted that this section was taken verbatim out of Appendix B of Reference 8. Derivations are made for both primed and unprimed dimensional derivatives for the three force and three moment equations. The linearization process is carried out for a specific set of steady-state conditions and is generally valid for small perturbations about this condition. For this study, the conditions selected about which to linearize the equations of motion for the F-16 were 0.6 Mach number at sea level. Data for this condition are given in Appendix A.

Aircraft Equations of Motion

Longitudinal Equations - Body Axes

$$F_{z_{cg}} = m\,(\dot{W} + pV - qU) - mg\cos\theta\cos\phi \qquad \text{(B-1)}$$

thus

$$\dot{W} = \frac{F_{z_{cg}}}{m} - pV + qU + g\cos\theta\cos\phi \qquad \text{(B-2)}$$

$$\frac{F_{z_{cg}}}{m} = \frac{\bar{q}s}{m}\left[C_{z_o} + C_{z_\alpha}\alpha + (C_{z_{\dot{\alpha}}}\dot{\alpha} + C_{z_q}q)\frac{c}{2V_T} + C_{z_u}\frac{\Delta U}{V_T} + C_{z_{\delta e}}\delta e + C_{z_{\delta f}}\delta f\right] \qquad \text{(B-3)}$$

Substituting Eq. (B-3) into (B-2) gives:

$$\dot{W} = \frac{\bar{q}s}{m}\left[C_{z_o} + C_{z_\alpha}\alpha + (C_{z_{\dot{\alpha}}}\dot{\alpha} + C_{z_q}q)\frac{c}{2V_T} + C_{z_u}\frac{\Delta U}{V_T} + C_{z_{\delta e}}\delta e + C_{z_{\delta f}}\delta f\right] - pV + qU + g\cos\theta\cos\phi \qquad \text{(B-4)}$$

To develop perturbation equations, a 1g wings level trim flight condition is examined where $\phi=0$, $\dot{\alpha}=0$, $q=0$, $\delta f=0$, $\Delta U=0$, $p=0$, and $\cos\theta$ is approximately one. The trim angle of attack and elevator position are α_T and δe_T respectively.

$$\dot{W} = 0 = \frac{\bar{q}S}{m}\left[C_{z_o} + C_{z_\alpha}\alpha_T + C_{z_{\delta e}}\delta e_T\right] + g \qquad \text{(B-5)}$$

Thus, the aerodynamic forces balance the vehicle's weight. To account for small variations from this trim condition, perturbation angle of attack α_p and elevator position δe_p are added to the equation. A term for small changes in sensed g is also included.

$$\dot{W} = \frac{\bar{q}S}{m}\left[C_{z_o} + C_{z_\alpha}(\alpha_T + \alpha_p) + C_{z_{\delta e}}(\delta e_T + \delta e_p)\right] + g - (g \sin\theta_T)\theta \qquad \text{(B-6)}$$

Cancelling the terms that are equal from Eq. (B-5) yields:

$$\dot{W} = \frac{\bar{q}S}{m}\left[C_{z_\alpha}\alpha_p + C_{z_{\delta e}}\delta e_p\right] - (g \sin\theta_T)\theta \qquad \text{(B-7)}$$

The equation is expanded to include perturbations in $\dot{\alpha}$, q, U and δf by referring to Eq. (B-4).

$$\dot{W} = \frac{\bar{q}S}{m}\left[C_{z_\alpha}\alpha_p + \{C_{z_{\dot{\alpha}}}\dot{\alpha} + C_{z_q}q\}\frac{c}{2V_T} + C_{z_u}\frac{\Delta U}{V_T} + C_{z_{\delta e}}\delta e_p + C_{z_{\delta f}}\delta f\right] + qU - (g \sin\theta_T)\theta \qquad \text{(B-8)}$$

The p subscript is dropped and $U = V_T$. ΔU is expressed as u. Thus, the perturbation equation is:

$$\dot{w} = (\frac{\bar{q}S}{m})C_{z_\alpha}\alpha + (\frac{\bar{q}Sc}{m2U})C_{z_{\dot{\alpha}}}\dot{\alpha} + (\frac{\bar{q}Sc}{m2U})C_{z_q}q + (\frac{\bar{q}S}{mU})C_{z_u}u$$

$$+ (\frac{\bar{q}S}{m})C_{z_{\delta e}}\delta e + (\frac{\bar{q}S}{m})C_{z_{\delta f}}\delta f + qU - (g \sin\theta_T)\theta \qquad \text{(B-9)}$$

or

$$\dot{w} = (Z_\alpha)\alpha + (Z_{\dot{\alpha}})\dot{\alpha} + (Z_q)q + (Z_u)u + (Z_{\delta e})\delta e$$

$$+ (Z_{\delta f})\delta f + qU - (g \sin\theta_T)\theta \qquad \text{(B-10)}$$

Dividing by U, letting $\dot{\alpha} = \frac{\dot{w}}{U}$, and gathering the $\dot{\alpha}$ terms on the left-hand side of the equation gives:

$$\dot{\alpha}(1 - \frac{Z_{\dot{\alpha}}}{U}) = (\frac{Z_\alpha}{U})\alpha + (\frac{Z_q}{U})q + (\frac{Z_u}{U})u + (\frac{Z_{\delta e}}{U})\delta e$$

$$+ (\frac{Z_{\delta f}}{U})\delta f + q - (\frac{g \sin\theta_T}{U})\theta \qquad \text{(B-11)}$$

$\frac{Z_{\dot{\alpha}}}{U}$ is very small and is ignored. Using the primed notation and noting that all states are perturbations from the trim condition, the equation can be expressed as:

$$\dot{\alpha} = (Z_\alpha')\alpha + (Z_q')q + (Z_u')u + (Z_{\delta e}')\delta e$$

$$+ (Z_{\delta f}')\delta f + (Z_\theta')\theta \qquad \text{(B-12)}$$

where $Z_\alpha' = \frac{Z_\alpha}{U} = \frac{\bar{q}S}{mU} C_{z_\alpha}$ (B-13)

$$Z_q' = 1 + \frac{Z_q}{U} = 1 + \frac{\bar{q}Sc}{m2U^2} C_{z_q} \qquad \text{(B-14)}$$

$$Z_u' = \frac{Z_u}{U} = \frac{\bar{q}S}{mU^2} C_{z_u} \qquad \text{(B-15)}$$

$$Z_{\delta e}' = \frac{Z_{\delta e}}{U} = \frac{\bar{q}S}{mU} C_{z_{\delta e}} \quad \text{(B-16)}$$

$$Z_{\delta f}' = \frac{Z_{\delta f}}{U} = \frac{\bar{q}S}{mU} C_{z_{\delta f}} \quad \text{(B-17)}$$

$$Z_{\theta}' = \frac{Z_{\theta}}{U} = -\frac{g}{U} \sin\theta_T \quad \text{(B-18)}$$

In a similar manner, the force equation in the x-axis is reduced to a perturbation equation.

$$F_{x_{cg}} = m(\dot{U} + qW - rV) + mg \sin\theta \quad \text{(B-19)}$$

thus

$$\dot{U} = \frac{F_{x_{cg}}}{m} - qW + rV - g \sin\theta \quad \text{(B-20)}$$

$$\frac{F_{x_{cg}}}{m} = \frac{\bar{q}S}{m}\left[C_{x_o} + C_{x_\alpha}\alpha + (C_{x_q}q)\frac{c}{2V_T} + C_{x_u}\frac{\Delta U}{V_T} + C_{x_{\delta e}}\delta e + C_{x_{\delta f}}\delta f\right] \quad \text{(B-21)}$$

$$\dot{U} = \frac{\bar{q}S}{m}\left[C_{x_o} + C_{x_\alpha}\alpha + (C_{x_q}q)\frac{c}{2V_T} + C_{x_u}\frac{\Delta U}{V_T} + C_{x_{\delta e}}\delta e + C_{x_{\delta f}}\delta f\right] - qW + rV - g \sin\theta \quad \text{(B-22)}$$

For trimmed flight, thrust exactly equals the drag forces.

$$T = \bar{q}S\left[C_{x_o} + C_{x_\alpha}\alpha_T + C_{x_{\delta e}}\delta e_T\right] \quad \text{(B-23)}$$

The perturbation equation is:

$$\dot{u} = \frac{\bar{q}S}{m} C_{x_\alpha} \alpha + \frac{\bar{q}Sc}{m2U} C_{x_q} q + \frac{\bar{q}S}{mU} C_{x_u} u + \frac{\bar{q}S}{m} C_{x_{\delta e}} \delta e$$

$$+ \frac{\bar{q}S}{m} C_{x_{\delta f}} \delta f - qW \frac{U}{U} + rV \frac{U}{U} - (g \cos\theta_T)\theta \quad \text{(B-24)}$$

By letting $\alpha_T = \frac{W}{U}$ and $\beta = \frac{V}{U}$, the equation can be written as:

$$\dot{u} = (X_\alpha)\alpha + (X_q)q + (X_u)u + (X_{\delta e})\delta e + (X_{\delta f})\delta f$$

$$- q\alpha_T U + r\beta U + X_\theta \theta \quad \text{(B-25)}$$

Assuming only longitudinal motion, β and r are zero and noting that all states are perturbations from trim conditions, the equation is expressed as:

$$\dot{u} = (X_\alpha')\alpha + (X_u')u + (X_{\delta e}')\delta e + (X_{\delta f}')\delta f$$

$$+ (X_q')q + (X_\theta')\theta \quad \text{(B-26)}$$

where $X_\alpha' = X_\alpha = \frac{\bar{q}S}{m} C_{x_\alpha}$ (B-27)

$$X_u' = X_u = \frac{\bar{q}S}{mU} C_{x_u} \quad \text{(B-28)}$$

$$X_{\delta e}' = X_{\delta e} = \frac{\bar{q}S}{m} C_{x_{\delta e}} \quad \text{(B-29)}$$

$$X_{\delta f} = X_{\delta f} = \frac{\bar{q}S}{m} C_{x_{\delta f}} \quad \text{(B-30)}$$

$$X_q' = X_q - U\alpha_T = \frac{\bar{q}Sc}{m2U} C_{x_q} - U\alpha_T \tag{B-31}$$

$$X_\theta' = X_\theta = -g \cos\theta_T \tag{B-32}$$

The pitching moment equation is used to develop the perturbation $\dot{q}$ equation.

$$M_y = \dot{q}\, I_{yy} + pr(I_{xx} - I_{zz}) - (r^2 - p^2)I_{xz} \tag{B-33}$$

$$\dot{q}\, I_{yy} = M_y - pr(I_{xx} - I_{zz}) + (r^2 - p^2)I_{xz} \tag{B-34}$$

For longitudinal motion only, r and p are zero and the equation becomes:

$$\dot{q} = \frac{M_y}{I_{yy}} \tag{B-35}$$

and

$$M_y = \bar{q}Sc \left[C_{m_o} + C_{m_\alpha}\alpha + (C_{m_{\dot{\alpha}}}\dot{\alpha} + C_{m_q}q) \frac{c}{2V_T} + C_{m_u}\frac{\Delta U}{V_T} + C_{m_{\delta e}}\delta e + C_{m_{\delta f}}\delta f\right]. \tag{B-36}$$

In trimmed flight, the moments are assumed to be zero.

$$M_y = \bar{q}Sc \left[C_{m_o} + C_{m_\alpha}\alpha_T + C_{m_{\delta e}}\delta e_T\right] = 0 \tag{B-37}$$

Letting $V_T = U$, introducing perturbation angle of attack and elevator position variables, and cancelling the above

terms that add to zero, gives the perturbation equation:

$$M_y = \bar{q}Sc\left[C_{m_\alpha}\alpha + \{C_{m_{\dot\alpha}}\dot\alpha + C_{m_q}q\}\frac{c}{2U} + \frac{C_{m_u}}{U}\Delta U + C_{m_{\delta e}}\delta e + C_{m_{\delta f}}\delta f\right] \quad \text{(B-38)}$$

Substituting Eq. (B-38) into (B-35)

$$\dot q = \frac{\bar{q}Sc}{I_{yy}}C_{m_\alpha}\alpha + \frac{\bar{q}Sc^2}{2UI_{yy}}C_{m_{\dot\alpha}}\dot\alpha + \frac{\bar{q}Sc^2}{2UI_{yy}}C_{m_q}q + \frac{\bar{q}Sc}{UI_{yy}}C_{m_u}\Delta U + \frac{\bar{q}Sc}{I_{yy}}C_{m_{\delta e}}\delta e + \frac{\bar{q}Sc}{I_{yy}}C_{m_{\delta f}}\delta f \quad \text{(B-39)}$$

In dimensional form, this is written as follows:

$$\dot q = (M_\alpha)\alpha + (M_{\dot\alpha})\dot\alpha + (M_q)q + (M_u)\Delta U + (M_{\delta e})\delta e + (M_{\delta f})\delta f \quad \text{(B-40)}$$

Substituting Eq (B-12) for $\dot\alpha$ into Eq. (B-40) and letting ΔU be represented by perturbation u yields:

$$\dot q = (M_\alpha + M_{\dot\alpha}Z_\alpha')\alpha + (M_q + M_{\dot\alpha}Z_q')q + (M_u + M_{\dot\alpha}Z_u')u + (M_{\delta e} + M_{\dot\alpha}Z_{\delta e}')\delta e + (M_{\delta f} + M_{\dot\alpha}Z_{\delta f}')\delta f + (M_{\dot\alpha}Z_\theta')\theta \quad \text{(B-41)}$$

Using the primed notation, the equation is represented as:

$$\dot q = (M_\alpha')\alpha + (M_q')q + (M_u')u + (M_{\delta e}')\delta e + (M_{\delta f}')\delta f + (M_\theta')\theta \quad \text{(B-42)}$$

where

$$M_\alpha' = \frac{\bar{q}Sc}{I_{yy}} (C_{m_\alpha})_b + \left[\frac{\bar{q}Sc^2}{2UI_{yy}} (C_{m_{\dot{\alpha}}})_b\right] Z_\alpha' \qquad \text{(B-43)}$$

$$M_q' = \frac{\bar{q}Sc^2}{2UI_{yy}} \left[(C_{m_q})_b + (C_{m_{\dot{\alpha}}})_b \, Z_q'\right] \qquad \text{(B-44)}$$

$$M_u' = \frac{\bar{q}Sc}{UI_{yy}} (C_{m_u})_b + \left[\frac{\bar{q}Sc^2}{2UI_{yy}} (C_{m_{\dot{\alpha}}})_b\right] Z_u' \qquad \text{(B-45)}$$

$$M_{\delta e}' = \frac{\bar{q}Sc}{I_{yy}} (C_{m_{\delta e}})_b + \left[\frac{\bar{q}Sc^2}{2UI_{yy}} (C_{m_{\dot{\alpha}}})_b\right] Z_{\delta e}' \qquad \text{(B-46)}$$

$$M_{\delta f}' = \frac{\bar{q}Sc}{I_{yy}} (C_{m_{\delta f}})_b + \left[\frac{\bar{q}Sc^2}{2UI_{yy}} (C_{m_{\dot{\alpha}}})_b\right] Z_{\delta f}' \qquad \text{(B-47)}$$

$$M_\theta' = \left[\frac{\bar{q}Sc^2}{2UI_{yy}} (C_{m_{\dot{\alpha}}})_b\right] Z_\theta' \qquad \text{(B-48)}$$

Note: $(\ )_b$ denotes coefficients that are expressed in the body axes.

Lateral-Directional Equation - Body Axes

The sideforce equation is:

$$F_{y_{cg}} = m(\dot{V} + rU - pW) - mg\cos\theta\sin\phi \qquad \text{(B-49)}$$

thus

$$\dot{V} = \frac{F_{y_{cg}}}{m} - rU + pW + g\cos\theta\sin\phi \qquad \text{(B-50)}$$

$$\frac{F_{y_{cg}}}{m} = \frac{\bar{q}S}{m}\left[C_{y_\beta}\beta + (C_{y_p}p + C_{y_r}r)\frac{b}{2V_T} + C_{y_{\delta a}}\delta a + C_{y_{\delta r}}\delta r + C_{y_{\delta c}}\delta c\right] \quad \text{(B-51)}$$

Substituting Eq.(B-51) into (B-50) and letting $U = V_T$ yields:

$$\dot{V} = \frac{\bar{q}S}{m}C_{y_\beta}\beta + \frac{\bar{q}Sb}{m2U}C_{y_p}p + \frac{\bar{q}Sb}{m2U}C_{y_r}r + \frac{\bar{q}S}{m}C_{y_{\delta a}}\delta a + \frac{\bar{q}S}{m}C_{y_{\delta r}}\delta r + \frac{\bar{q}S}{m}C_{y_{\delta c}}\delta c - rU + pW + g\cos\theta\sin\phi \quad \text{(B-52)}$$

Written with dimensional derivatives, this becomes:

$$\dot{V} = (Y_\beta)\beta + (Y_p)p + (Y_r)r + (Y_{\delta a})\delta a + (Y_{\delta r})\delta r + (Y_{\delta c})\delta c - rU + pW + g\cos\theta\sin\phi \quad \text{(B-53)}$$

Dividing by U, letting $\dot{\beta} = \frac{\dot{V}}{U}$, $\beta = \frac{V}{U}$, $\alpha = \frac{W}{U}$, $\sin\phi = \phi$ in radians, and gathering terms together yields:

$$\dot{\beta} = \left(\frac{Y_\beta}{U}\right)\beta + \left(\frac{Y_p}{U} + \alpha\right)p + \left(\frac{Y_r}{U} - 1\right)r + \left(\frac{Y_{\delta a}}{U}\right)\delta a + \left(\frac{Y_{\delta r}}{U}\right)\delta r + \left(\frac{Y_{\delta c}}{U}\right)\delta c + \left(\frac{g\cos\theta_T}{U}\right)\phi \quad \text{(B-54)}$$

Using the primed notation, the equation is represented as:

$$\dot{\beta} = (Y_\beta')\beta + (Y_p')p + (Y_r')r + (Y_{\delta a}')\delta a + (Y_{\delta r}')\delta r + (Y_{\delta c}')\delta c + (Y\phi')\phi \quad \text{(B-55)}$$

where

$$Y_\beta' = \frac{\bar{q}S}{mU} (C_{y_\beta})_b \tag{B-56}$$

$$Y_p' = \frac{\bar{q}Sb}{m2U^2} (C_{y_p})_b + \alpha_T \tag{B-57}$$

$$Y_r' = \frac{\bar{q}Sb}{m2U^2} (C_{y_r})_b - 1 \tag{B-58}$$

$$Y_{\delta a}' = \frac{\bar{q}S}{mU} (C_{y_{\delta a}})_b \tag{B-59}$$

$$Y_{\delta r}' = \frac{\bar{q}S}{mU} (C_{y_{\delta r}})_b \tag{B-60}$$

$$Y_{\delta c}' = \frac{\bar{q}S}{mU} (C_{y_{\delta c}})_b \tag{B-61}$$

$$Y_\phi' = \frac{g \cos\theta_T}{U} \tag{B-62}$$

The yawing moment equation can be expressed as:

$$M_z = \dot{r} I_{zz} + qp(I_{yy} - I_{xx}) - (\dot{p} - qr)I_{xz} \tag{B-63}$$

Assuming $q = 0$, this reduces to:

$$\dot{r} I_{zz} = M_z + \dot{p} I_{xz} \tag{B-64}$$

$$M_z = \bar{q}Sb \left[C_{n_\beta}\beta + \{C_{n_p} p + C_{n_r} r\} \frac{b}{2V_T} + C_{n_{\delta a}}\delta a + C_{n_{\delta r}}\delta r + C_{n_{\delta c}}\delta c\right] \tag{B-65}$$

Combining Eqs. (B-64) and (B-65) and solving for $\dot{r}$ with $U = V_T$ gives:

$$\dot{r} = \frac{\bar{q}Sb}{I_{zz}} C_{n_\beta}\beta + \frac{\bar{q}Sb^2}{I_{zz}2U} C_{n_p} p + \frac{\bar{q}Sb^2}{I_{zz}2U} C_{n_r} r + \frac{\bar{q}Sb}{I_{zz}} C_{n_{\delta a}}\delta a$$

$$+ \frac{\bar{q}Sb}{I_{zz}} C_{n_{\delta r}}\delta r + \frac{\bar{q}Sb}{I_{zz}} C_{n_{\delta c}}\delta c + \dot{p}\frac{I_{xz}}{I_{zz}} \quad \text{(B-66)}$$

In dimensional derivative notation, this is:

$$\dot{r} = (N_\beta)\beta + (N_p)p + (N_r)r + (N_{\delta a})\delta a + (N_{\delta r})\delta r$$

$$+ (N_{\delta c})\delta c + \dot{p}\frac{I_{xz}}{I_{zz}} \quad \text{(B-67)}$$

The rolling moment equation is written as:

$$M_x = \dot{p}\, I_{xx} + qr(I_{zz} - I_{yy}) - (pq + \dot{r})I_{xz} \quad \text{(B-68)}$$

Assuming $q = 0$, the equation reduces to:

$$\dot{p}\, I_{xx} = M_x + \dot{r}\, I_{xz} \quad \text{(B-69)}$$

$$M_x = \bar{q}Sb \left[C_{\ell_\beta}\beta + (C_{\ell_p} p + C_{\ell_r} r)\frac{b}{2V_T} + C_{\ell_{\delta a}}\delta a \right.$$

$$\left. + C_{\ell_{\delta r}}\delta r + C_{\ell_{\delta c}}\delta_c \right] \quad \text{(B-70)}$$

Combining Eqs. (B-69) and (B-70) and solving for $\dot{p}$ with $U = V_T$ gives:

$$\dot{p} = \frac{\bar{q}Sb}{I_{xx}} C_{\ell_\beta}\beta + \frac{\bar{q}Sb^2}{I_{xx}2U} C_{\ell_p} p + \frac{\bar{q}Sb^2}{I_{xx}2U} C_{\ell_r} r + \frac{\bar{q}Sb}{I_{xx}} C_{\ell_{\delta a}}\delta a$$

$$+ \frac{\bar{q}Sb}{I_{xx}} C_{\ell_{\delta r}}\delta r + \frac{\bar{q}Sb}{I_{xx}} C_{\ell_{\delta c}}\delta c + \dot{r}\frac{I_{xz}}{I_{xx}} \quad \text{(B-71)}$$

$$\dot{p} = (L_\beta)\beta + (L_p)p + (L_r)r + (L_{\delta a})\delta a + (L_{\delta r})\delta r$$

$$+ (L_{\delta c})\delta c + \dot{r}\frac{I_{xz}}{I_{xx}} \quad \text{(B-72)}$$

Equations (B-67) and (B-72) are solved to give expressions for $\dot{r}$ and $\dot{p}$. Written in primed derivatives, these are:

$$\dot{r} = (N_\beta')\beta + (N_p')p + (N_r')r + (N_{\delta a}')\delta a$$

$$+ (N_{\delta r}')\delta r + (N_{\delta c}')\delta c \quad \text{(B-73)}$$

where

$$N_i' = \frac{N_i + \frac{I_{xz}}{I_{zz}} L_i}{1 - \frac{I_{xz}^2}{(I_{xx})(I_{zz})}} \quad \text{for } i = \beta,\ p,\ r,\ \delta a,\ \delta r,\ \delta c \quad \text{(B-74)}$$

and

$$\dot{p} = (L_\beta')\beta + (L_p')p + (L_r')r + (L_{\delta a}')\delta a$$

$$+ (L_{\delta r}')\delta r + (L_{\delta c}')\delta c \quad \text{(B-75)}$$

where

$$L_i' = \frac{L_i + \frac{I_{xz}}{I_{xx}} N_i}{1 + \frac{(I_{xz})^2}{(I_{xx})(I_{zz})}} \quad \text{for } i = \beta,\ p,\ r,\ \delta a,\ \delta r,\ \delta c \tag{B-76}$$

The State Equations

Equations (B-12), (B-26) and (B-42) are combined with an expression for $\dot{\theta}$ and first-order actuator models (developed in Chapter II) to form the longitudinal state equations.

$$\dot{\theta} = q \cos\phi - r \sin\phi \tag{B-77}$$

Assuming ϕ is small and r is zero, this becomes:

$$\dot{\theta} = q \tag{B-78}$$

Thus, the longitudinal state equations are:

$$\begin{bmatrix} \dot{\theta} \\ \dot{u} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\delta e} \\ \dot{\delta f} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ X_\theta' & X_u' & X_\alpha' & X_q' & X_{\delta e}' & X_{\delta f}' \\ Z_\theta' & Z_u' & Z_\alpha' & Z_q' & Z_{\delta e}' & Z_{\delta f}' \\ M_\theta' & M_u' & M_\alpha' & M_q' & M_{\delta e}' & M_{\delta f}' \\ 0 & 0 & 0 & 0 & -20 & 0 \\ 0 & 0 & 0 & 0 & 0 & -20 \end{bmatrix} \begin{bmatrix} \theta \\ u \\ \alpha \\ q \\ \delta e \\ \delta f \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 20 & 0 \\ 0 & 20 \end{bmatrix} \begin{bmatrix} \delta e_{cmd} \\ \delta f_{cmd} \end{bmatrix} \tag{B-79}$$

Units are radians, feet per second, and radians per second.

Equations (B-55), (B-73) and (B-75) are combined with an expression for $\dot{\phi}$ and first-order actuator models to form the lateral-directional state equations.

$$\dot{\phi} = p + q \sin\phi \tan\theta + r \cos\phi \tan\theta \qquad \text{(B-80)}$$

Assuming $\theta = 0$, this becomes:

$$\dot{\phi} = p \qquad \text{(B-81)}$$

Thus, the lateral-directional state equations are:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\beta} \\ \dot{p} \\ \dot{r} \\ \dot{\delta a} \\ \dot{\delta r} \\ \dot{\delta c} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ Y_{\phi}' & Y_{\beta}' & Y_p' & Y_r' & Y_{\delta a}' & Y_{\delta r}' & Y_{\delta c}' \\ 0 & L_{\beta}' & L_p' & L_r' & L_{\delta a}' & L_{\delta r}' & L_{\delta c}' \\ 0 & N_{\beta}' & N_p' & N_r' & N_{\delta a}' & N_{\delta r}' & N_{\delta c}' \\ 0 & 0 & 0 & 0 & -20 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -20 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -20 \end{bmatrix} \begin{bmatrix} \phi \\ \beta \\ p \\ r \\ \delta a \\ \delta r \\ \delta c \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 20 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 20 \end{bmatrix} \begin{bmatrix} \delta a_{cmd} \\ \delta r_{cmd} \\ \delta c_{cmd} \end{bmatrix}$$

(B-82)

These state equations must still be transformed as shown in Chapter II to obtain longitudinal and lateral accelerations as states. The stability axes coefficients must be converted to body axes coefficients for use in the equations previously developed. The conversion equations are:

$$C_{Z_\alpha} = (-C_{L_\alpha} - C_D) \cos^2\alpha_T + (-C_{D_u} - 2C_D) \sin^2\alpha_T$$
$$+ (-C_{L_u} - C_L - C_{D_\alpha}) \cos\alpha_T \sin\alpha_T \qquad \text{(B-83)}$$

$$C_{Z_{\dot{\alpha}}} = -C_{L_{\dot{\alpha}}} \cos^2\alpha_T \tag{B-83a}$$

$$C_{Z_q} = -C_{L_q} \cos\alpha_T \tag{B-84}$$

$$C_{Z_u} = (-C_{L_u} - 2C_L) \cos^2\alpha_T + (C_{D_\alpha} - C_L) \sin^2\alpha_T$$

$$+ (C_{L_\alpha} - C_{D_u} - C_D) \cos\alpha_T \sin\alpha_T \tag{B-85}$$

$$C_{Z_\delta} = -C_{L_\delta} \cos\alpha_T - C_{D_\delta} \sin\alpha_T \tag{B-86}$$

$$C_{X_\alpha} = (-C_{D_\alpha} + C_L) \cos^2\alpha_T + (C_{L_u} + 2C_L) \sin^2\alpha_T$$

$$+ (-C_{D_u} - C_D + C_{L_\alpha}) \cos\alpha_T \sin\alpha_T \tag{B-87}$$

$$C_{X_q} = C_{L_q} \sin\alpha_T \tag{B-88}$$

$$C_{X_u} = (-C_{D_u} - 2C_D) \cos^2\alpha_T + (-C_{L_\alpha} - C_D) \sin^2\alpha_T$$

$$+ (C_{D_\alpha} + C_{L_u} + C_L) \cos\alpha_T \sin\alpha_T \tag{B-89}$$

$$C_{X_\delta} = -C_{D_\delta} \cos\alpha_T + C_{L_\delta} \sin\alpha_T \tag{B-90}$$

$$(C_{M_\alpha})_b = C_{M_\alpha} \cos\alpha_T + (C_{M_u} + 2C_M) \sin\alpha_T \tag{B-91}$$

$$(C_{M_{\dot{\alpha}}})_b = C_{M_{\dot{\alpha}}} \cos\alpha_T \tag{B-92}$$

$$(C_{M_u})_b = (C_{M_u} + 2C_M) \cos\alpha_T - C_{M_\alpha} \sin\alpha_T \tag{B-93}$$

$$(C_{M_q})_b = C_{M_q} \tag{B-94}$$

$$(C_{M_\delta})_b = C_{M_\delta} \tag{B-95}$$

where $(\ )_b$ is used to distinguish body axes from stability axes when necessary.

The equations for converting the lateral derivatives to body axes are:

$$(C_{\ell_\beta})_b = C_{\ell_\beta} \cos\alpha_T - C_{n_\beta} \sin\alpha_T \tag{B-96}$$

$$(C_{\ell_p})_b = C_{\ell_p} \cos^2\alpha_T + C_{n_r} \sin^2\alpha_T - (C_{\ell_r}+C_{n_p}) \sin\alpha_T \cos\alpha_T \tag{B-97}$$

$$(C_{\ell_r})_b = C_{\ell_r} \cos^2\alpha_T - (C_{n_r}-C_{\ell_p}) \sin\alpha_T \cos\alpha_T - C_{n_p} \sin^2\alpha_T \tag{B-98}$$

$$(C_{\ell_\delta})_b = C_{\ell_\delta} \cos\alpha_T - C_{n_\delta} \sin\alpha_T \tag{B-99}$$

$$(C_{n_\beta})_b = C_{n_\beta} \cos\alpha_T + C_{\ell_\beta} \sin\alpha_T \tag{B-100}$$

$$(C_{n_p})_b = C_{n_p} \cos^2\alpha_T - (C_{n_r}-C_{\ell_p}) \sin\alpha_T \cos\alpha_T - C_{\ell_r} \sin^2\alpha_T \tag{B-101}$$

$$(C_{n_r})_b = C_{n_r} \cos^2\alpha_T + (C_{\ell_r}+C_{n_p}) \sin\alpha_T \cos\alpha_T + C_{\ell_p} \sin^2\alpha_T \qquad \text{(B-102)}$$

$$(C_{n_\delta})_b = C_{n_\delta} \cos\alpha_T + C_{\ell_\delta} \sin\alpha_T \qquad \text{(B-103)}$$

$$(C_{y_\beta})_b = C_{y_\beta} \qquad \text{(B-104)}$$

$$(C_{y_p})_b = C_{y_p} \cos\alpha_T - C_{y_r} \sin\alpha_T \qquad \text{(B-105)}$$

$$(C_{y_r})_b = C_{y_r} \cos\alpha_T + C_{y_p} \sin\alpha_T \qquad \text{(B-106)}$$

$$(C_{y_\delta})_b = C_{y_\delta} \qquad \text{(B-107)}$$

All of the computations to develop the body axes primed derivatives from stability axes coefficients are performed by the CAT program (see Appendix D).

Miscellaneous Equations

To convert inertias from the body axes to the stability axes, the following equations are used.

$$(I_{xx})_S = (I_{xx})_B \cos^2\alpha_T + (I_{zz})_B \sin^2\alpha_T - 2(I_{xz})_B \cos\alpha_T \sin\alpha_T \qquad \text{(B-108)}$$

$$(I_{zz})_S = (I_{xx})_B \sin^2\alpha_T + (I_{zz})_B \cos^2\alpha_T + 2(I_{xz})_B \cos\alpha_T \sin\alpha_T \qquad \text{(B-109)}$$

$$(I_{xz})_S = [(I_{xx})_B - (I_{zz})_B] \cos\alpha_T \sin\alpha_T + (I_{xz})_B [\cos^2\alpha_T - \sin^2\alpha_T] \quad \text{(B-110)}$$

where $(\ )_S$ is used to denote the stability axes.

Accelerations at points other than the center of gravity are calculated using:

$$A_x = A_{x_{cg}} - \left(\frac{\ell_x}{1845}\right)\left(\frac{q^2 + r^2}{57.3}\right) + \left(\frac{\ell_y}{1845}\right)\left(\frac{pq}{57.3} - \dot{r}\right) + \left(\frac{\ell_z}{1845}\right)\left(\frac{pr}{57.3} + \dot{q}\right) \quad \text{(B-111)}$$

$$A_y = A_{y_{cg}} + \left(\frac{\ell_x}{1845}\right)\left(\frac{pq}{57.3} + \dot{r}\right) - \left(\frac{\ell_y}{1845}\right)\left(\frac{p^2 + r^2}{57.3}\right) + \left(\frac{\ell_z}{1845}\right)\left(\frac{qr}{57.3} - \dot{p}\right) \quad \text{(B-112)}$$

$$A_n = A_{n_{cg}} - \left(\frac{\ell_x}{1845}\right)\left(\frac{pr}{57.3} - \dot{q}\right) - \left(\frac{\ell_y}{1845}\right)\left(\frac{rq}{57.3} + \dot{p}\right) + \left(\frac{\ell_z}{1845}\right)\left(\frac{p^2 + q^2}{57.3}\right) \quad \text{(B-113)}$$

Accelerations are in units of g, angular rates are in units of degrees per second, and angular accelerations are in degrees per second^2. The distances ℓ_x, ℓ_y, ℓ_z are measured in feet. The ℓ_x distance is positive moving forward from the CG along the x-axis. The ℓ_y distance is positive along the y-axis moving out the right wing from the center of gravity. The ℓ_z distance is positive along the z-axis or down from the CG.

Angle of attack and angle of sideslip are expressed as:

$$\alpha = \tan^{-1}\left(\frac{W}{U}\right) \qquad \text{and} \qquad \beta = \sin^{-1}\left(\frac{V}{V_T}\right) \tag{B-114}$$

where $V_T = (U^2 + V^2 + W^2)^{\frac{1}{2}}$ (B-115)

Appendix E: State-Space Control System Matrices

This appendix contains the matrices that were used to construct the F-16 state-space system. The aircraft longitudinal states were incremental forward velocity, perturbation angle of attack, pitch angle, pitch rate, horizontal tail deflection, and altitude above mean sea level:

$$\underline{x} = [\, u \quad \alpha \quad \theta \quad q \quad \delta_{HT} \quad h_{msl} \,]^T$$

The open loop longitudinal state-space matrix is represented in the form of

$$\begin{aligned} \underline{x} &= A\underline{x} + Bu \\ \underline{y} &= C\underline{x} + Du \end{aligned} \qquad \text{(E.1)}$$

where

$$A = \begin{bmatrix} -0.0148 & 0.6524 & -0.5618 & -0.3132 & 0.1225 & 0.0000 \\ -0.0048 & -1.4921 & -0.0013 & 0.9928 & -0.1882 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 \\ -0.0206 & 9.7532 & 0.0003 & -0.9591 & -19.0410 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & -20.0000 & 0.0000 \\ 0.0000 & -11.6928 & 11.6928 & 0.0000 & 0.0000 & 0.0000 \end{bmatrix}$$

$$B = \begin{bmatrix} 0. \\ 0. \\ 0. \\ 0. \\ 20. \\ 0. \end{bmatrix}$$

$$C = \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 \\ 0.0016 & 0.6155 & 0.0005 & -0.0046 & -0.0754 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$$

$$D = \begin{bmatrix} 0. \\ 0. \\ 0. \\ 0. \end{bmatrix}$$

The state-space system that represents the longitudinal feedback paths is written in the same form as Eq (E.1) with the matrices A_k, B_k, C_k, and D_k :

A_k =

0.	1.	0.	0.	0.
-12.	-13.	0.	0.	0.
0.	0.	0.	1.	0.
0.	0.	0.	-12.	0.
0.	0.	0.	0.	-10.

B_k =

0.	0.	0.
1.	0.	0.
0.	0.	0.
0.	1.	0.
0.	0.	1.

C_k =

-8.6080	4.3040	-64.4400	9.6660	-5.0000

D_k =

-1.0760	-3.2220	0.0000

The longitudinal feedforward path is represented by A_E, B_E, C_E, and D_E :

$$A_E = \begin{matrix} 0 & 1 \\ 0 & -60 \end{matrix} \qquad B_E = \begin{matrix} 0 \\ 1 \end{matrix}$$

$$C_E = \begin{matrix} -117 & -23.4 \end{matrix} \qquad D_E = 0$$

The entire aircraft plant, both longitudinal and lateral-directional, is shown on the following page, again in the same format as Eq (E.1) with the matrices A, B, C, D.

A =

Columns 1 thru 8

-0.0148	0.0000	0.6524	0.0000	-0.5618	0.0000	0.0000	-0.3132
0.0000	-0.4389	0.0000	0.0000	0.0000	0.0480	0.0273	0.0000
-0.0048	0.0000	-1.4921	0.0000	-0.0013	0.0000	0.0000	0.3928
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
0.0000	-45.0950	0.0000	0.0000	0.0000	0.0000	-2.4690	0.0000
-0.0206	0.0000	9.7532	0.0000	0.0003	0.0000	0.0000	-0.9591
0.0000	7.4884	0.0000	0.0000	0.0000	0.0000	-0.0081	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	-11.6928	0.0000	11.6928	0.0000	0.0000	0.0000
0.0560	0.0000	17.4469	0.0000	0.0152	0.0000	0.0000	0.0844

Columns 9 thru 14

0.0000	0.1225	0.0000	0.0000	0.0000	0.0000
-0.9956	0.0000	0.0047	0.0612	0.0000	0.0000
0.0000	-0.1882	0.0000	0.0000	0.0000	0.0000
1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
-0.0066	0.0000	-64.5965	11.3885	0.0000	0.0000
0.0000	-19.0410	0.0000	0.0000	0.0000	0.0000
-0.8294	0.0000	-2.2082	-5.9140	0.0000	0.0000
0.0000	-20.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	-20.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	-20.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	2.2002	0.0000	0.0000	0.0000	0.0000

B =

0.	0.	0.
0.	0.	0.
0.	0.	0.
0.	0.	0.
0.	0.	0.
0.	0.	0.
0.	0.	0.
0.	0.	0.
0.	0.	0.
20.	0.	0.
0.	20.	0.
0.	0.	20.
0.	0.	0.
0.	0.	0.

C =

Columns 1 thru 8

0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
0.0016	0.0000	0.6155	0.0000	0.0005	0.0000	0.0000	-0.0046
0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	-0.1028	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Columns 9 thru 14

0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	-0.0754	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
-0.0047	0.0000	-0.0149	-0.0224	0.0000	0.0000

D = 0

The states in the aircraft plant are incremental forward velocity, sideslip angle, perturbation angle of attack, heading angle, pitch angle, bank angle, roll rate, pitch rate, yaw rate, horizontal tail deflection, flaperon deflection, rudder deflection, altitude, and altitude rate:

$$\underline{x} = [\, u \;\; \beta \;\; \alpha \;\; \psi \;\; \theta \;\; \phi \;\; p \;\; q \;\; r \;\; \delta_{HT} \;\; \delta_{F} \;\; \delta_{R} \;\; h_{msl} \;\; h_{msl} \,]^T$$

The complete feedback state-space system is represented by the matrices A_k, B_k, C_k, and D_k. The states in this case are 'fictitious'.

AK =

$$\begin{matrix}
0. & 1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\
-12. & -13. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\
0. & 0. & 0. & 1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\
0. & 0. & 0. & -12. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\
0. & 0. & 0. & 0. & -10. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\
0. & 0. & 0. & 0. & 0. & -50. & 0. & 0. & 0. & 0. & 0. & 0. \\
0. & 0. & 0. & 0. & 0. & 0. & 0. & 1. & 0. & 0. & 0. & 0. \\
0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 1. & 0. & 0. & 0. \\
0. & 0. & 0. & 0. & 0. & 0. & -750. & -815. & -66. & 0. & 0. & 0. \\
0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 1. & 0. \\
0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 1. \\
0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & -750. & -815. & -66.
\end{matrix}$$

BK =

$$\begin{matrix}
0. & 0. & 0. & 0. & 0. & 0. \\
1. & 0. & 0. & 0. & 0. & 0. \\
0. & 0. & 0. & 0. & 0. & 0. \\
0. & 1. & 0. & 0. & 0. & 0. \\
0. & 0. & 1. & 0. & 0. & 0. \\
0. & 0. & 0. & 1. & 0. & 0. \\
0. & 0. & 0. & 0. & 0. & 0. \\
0. & 0. & 0. & 0. & 0. & 0. \\
0. & 0. & 0. & 1. & 0. & 0. \\
0. & 0. & 0. & 0. & 0. & 0. \\
0. & 0. & 0. & 0. & 0. & 0. \\
0. & 0. & 0. & 0. & 1. & 0.
\end{matrix}$$

CK =

Columns 1 thru 8

$$\begin{matrix}
-8.6080 & 4.3040 & -64.4400 & 9.6660 & -5.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -6.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -3.3174 & 11.6489
\end{matrix}$$

Columns 9 thru 12

$$\begin{matrix}
0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 \\
2.8163 & 0.0000 & -562.5000 & -112.5000
\end{matrix}$$

DK =

$$\begin{matrix}
-1.0760 & -3.2220 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -9.6600
\end{matrix}$$

The feedforward system for the complete system can be represented by the matrices A_E , B_E , C_E , and D_E , with D_E being identically equal to zero:

```
AE       =

     0.     1.     0.     0.
     0.   -60.     0.     0.
     0.     0.   -50.     0.
     0.     0.     0.   -50.

BE       =

     0.     0.
     1.     0.
     0.     1.
     0.     1.

CE       =

 -117.0000  -23.4000    0.0000    0.0000
    0.0000    0.0000   -6.0000    0.0000
    0.0000    0.0000    0.0000   -0.2216
```

The closed loop state-space system for the aircraft can now be created using the three state-space sytsems described above and the derivation presented in Chapter II of this thesis. Because of the size of the closed loop system, the representative matrices are presented on the proceeding pages. The closed loop system is in the form of Eq (E.1) with the matrices being A_{CL} , B_{CL} , C_{CL} , and D_{CL} , with D_{CL} being identically equal to zero.

A =

1.0D+04 *

Columns 1 thru 8

0.0000	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	-0.0060	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	-0.0050	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	-0.0050	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-0.0001	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	-0.0045	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
-0.2340	-0.0468	0.0000	0.0000	0.0000	0.0000	0.0040	0.0000
0.0000	0.0000	-0.0120	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	-0.0004	0.0000	-0.0020	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-0.0012	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0017	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Columns 9 thru 16

0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
-0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	-0.0001	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	-0.0002	0.0000	0.0000	0.0000	-0.0065	0.0011
0.0000	0.0000	0.0000	-0.0001	0.0000	-0.0019	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	-0.0001	0.0000	-0.0002	-0.0006
0.0000	0.0000	0.0000	0.0021	0.0000	-0.0025	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-0.0020	0.0000
0.0000	0.0000	0.0000	0.0000	-0.0001	0.0000	-0.0003	-0.0024
0.0012	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0002	0.0000	0.0000

0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000

Columns 17 thru 24

0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0172	-0.0086	0.1289	-0.0193	0.0100	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0120
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	-0.0012	-0.0013	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	-0.0012	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-0.0010	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-0.0050
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Columns 25 thru 30

0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0066	-0.0233	-0.0056	0.0000	1.1250	0.2250
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0001	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0001	0.0000	0.0000	0.0000
-0.0750	-0.0815	-0.0066	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0001	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0001
0.0000	0.0000	0.0000	-0.0750	-0.0815	-0.0066

```
B     =

     0.     0.
     1.     0.
     0.     1.
     0.     1.
     0.     0.
     0.     0.
     0.     0.
     0.     0.
     0.     0.
     0.     0.
     0.     0.
     0.     0.
     0.     0.
     0.     0.
     0.     0.
     0.     0.
     0.     0.
     0.     0.
     0.     0.
     0.     0.
     0.     0.
     0.     0.
     0.     0.
     0.     0.
     0.     0.
     0.     0.
     0.     0.
     0.     0.
     0.     0.
     0.     0.


C     =

  Columns 1 thru 8
  0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
  0.0000   0.0000   0.0000   0.0000   0.0016   0.0000   0.6155   0.0000
  0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   1.0000   0.0000
  0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
  0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
  0.0000   0.0000   0.0000   0.0000   0.0000  -0.1028   0.0000   0.0000
  Columns 9 thru 16
  0.0000   0.0000   0.0000   1.0000   0.0000   0.0000   0.0000   0.0000
  0.0005   0.0000   0.0000  -0.0046   0.0000  -0.0754   0.0000   0.0000
  0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
  0.0000   0.0000   1.0000   0.0000   0.0000   0.0000   0.0000   0.0000
  0.0000   0.0000   0.0000   0.0000   1.0000   0.0000   0.0000   0.0000
  0.0000   0.0000   0.0000   0.0000  -0.0047   0.0000  -0.0149  -0.0224
  Columns 17 thru 24
  0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
  0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
  0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
  0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
  0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
  0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
  Columns 25 thru 30
  0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
  0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
  0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
  0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
  0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
  0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
```

The eigenvalues, or poles, of the open loop plant are

```
  0.0000 + 0.0000i
  0.0000 + 0.0000i
  0.0000 + 0.0000i
 -0.0086 + 0.0719i
 -0.0086 - 0.0719i
 -0.0822 - 0.0000i
  1.9006 - 0.0000i
 -2.4504 + 0.0000i
 -0.6024 + 2.9269i
 -0.6024 - 2.9269i
 -4.3494 + 0.0000i
-20.0000 + 0.0000i
-20.0000 + 0.0000i
-20.0000 + 0.0000i
```

The eigenvalues of the closed loop system are given by

```
  0.0000 + 0.0000i
  0.0000 + 0.0000i
  0.0000 + 0.0000i
  0.0000 + 0.0000i
 -0.0002 + 0.0000i
 -0.0123 + 0.0000i
 -0.0149 + 0.0000i
 -0.6415 + 0.0000i
 -1.0000 + 0.0000i
 -1.3308 + 0.0000i
 -2.1112 + 0.0000i
 -1.4835 - 2.2187i
 -1.4835 + 2.2187i
 -3.3356 + 3.1843i
 -3.3356 - 3.1843i
-10.2819 + 0.0000i
-12.0000 - 0.0000i
 -9.0094 -10.4668i
 -9.0094 +10.4668i
-15.0000 + 0.0000i
-14.3102 -16.4349i
-14.3102 +16.4349i
-15.3023 -15.6413i
-15.3023 +15.6413i
-50.0000 - 0.0000i
-50.0000 + 0.0000i
-50.0000 + 0.0000i
-54.4802 - 0.0000i
-58.6432 - 0.0000i
-60.0000 + 0.0000i
```

Appendix F: Root Locus Plots From Development of Altitude Controller

Appendix F contains the root locus plots that were used to design the compensators of the altitude controller for the terrain avoidance system. The root locus plot is a plot of the control system's characteristic equation and shows the migration of the open loop poles to the open loop zeros as system gain is increased, hence, the open loop transfer function is used. The characteristic equation is given by

$$1 + GH = 0 \tag{F.1}$$

or

$$GH = -1 \tag{F.2}$$

where G is the plant of the system and H is the compensator, which in this case is in the feedforward path.

From Eq (F.2), two conditions for magnitude and angle must be satisfied for the roots of the transfer function to lie on a branch of the root locus:

$$|GH| = 1 \tag{F.3}$$

$$GH = 180 \tag{F.4}$$

For the controller designed in this thesis, lead compensators were used to obtain the desired system response. A lead compensator takes the form of

$$H = (s + a)/(s+b) \tag{F.5}$$

where $a < b$

The compensator pole and zero are placed so that the root locus will pass through the location of the desired closed loop poles. Knowing the location of the desired poles, Eq

(F.4) can be used to design the compensator. For a lead compensator, the zero location is usually chosen, and then Eq (F.4) is used to determine the location of the pole. A lead compensator in the forward path will tend to pull the branches of a root locus further over into the left-half plane, while a lag compensator will have the opposite effect. Once the locus passes through the desired poles, the compensator gain is adjusted until the desired poles are reached. The location of these poles will become the poles of the closed loop system.

The design method discussed above was used in the design of the altitude controller for this thesis. A more detailed discussion of the compensators chosen for this design is given in Chapter III.

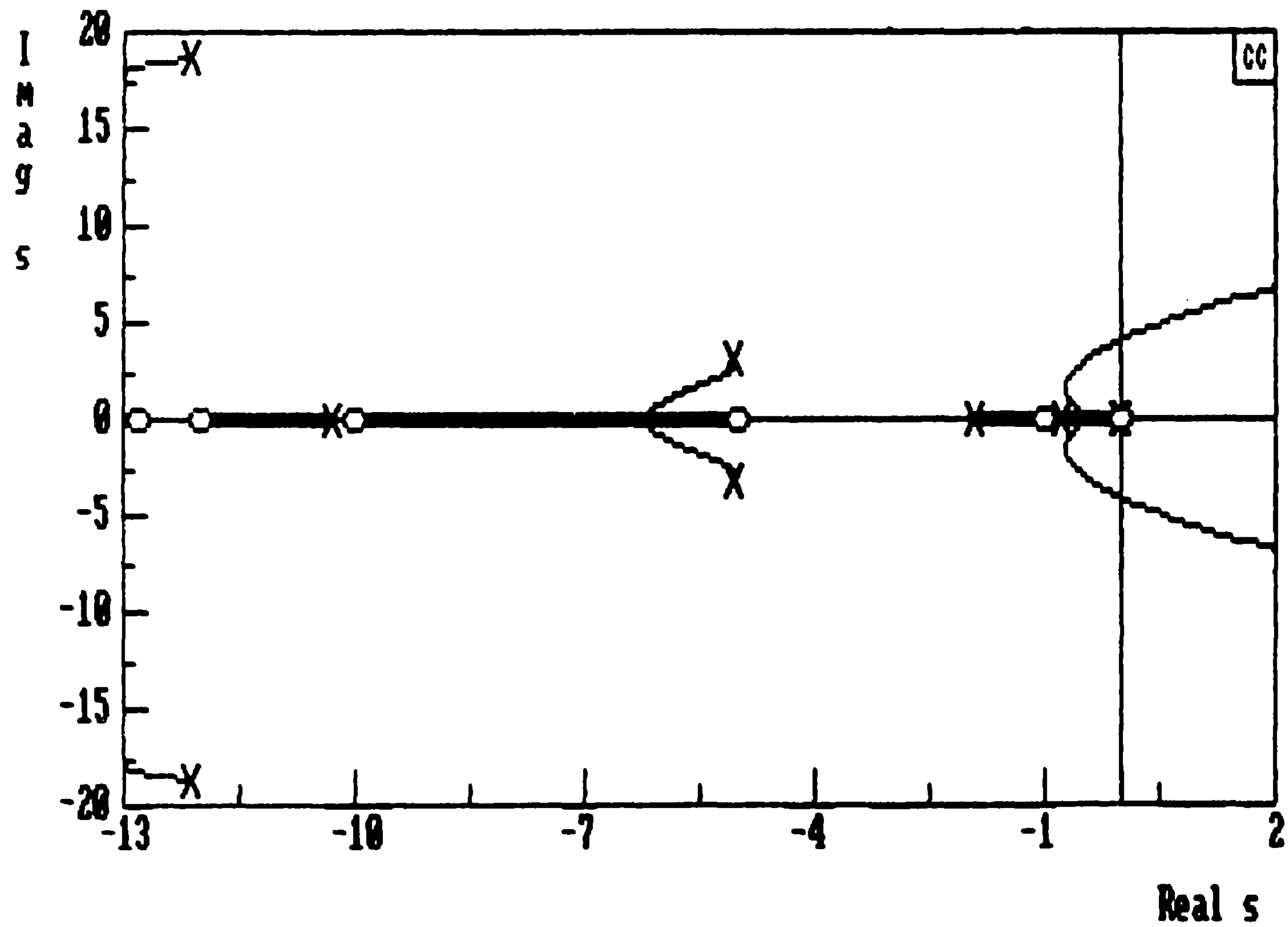


Figure F.1: Root Locus of Flight Path Angle to Pitch Rate Command Without Compensation

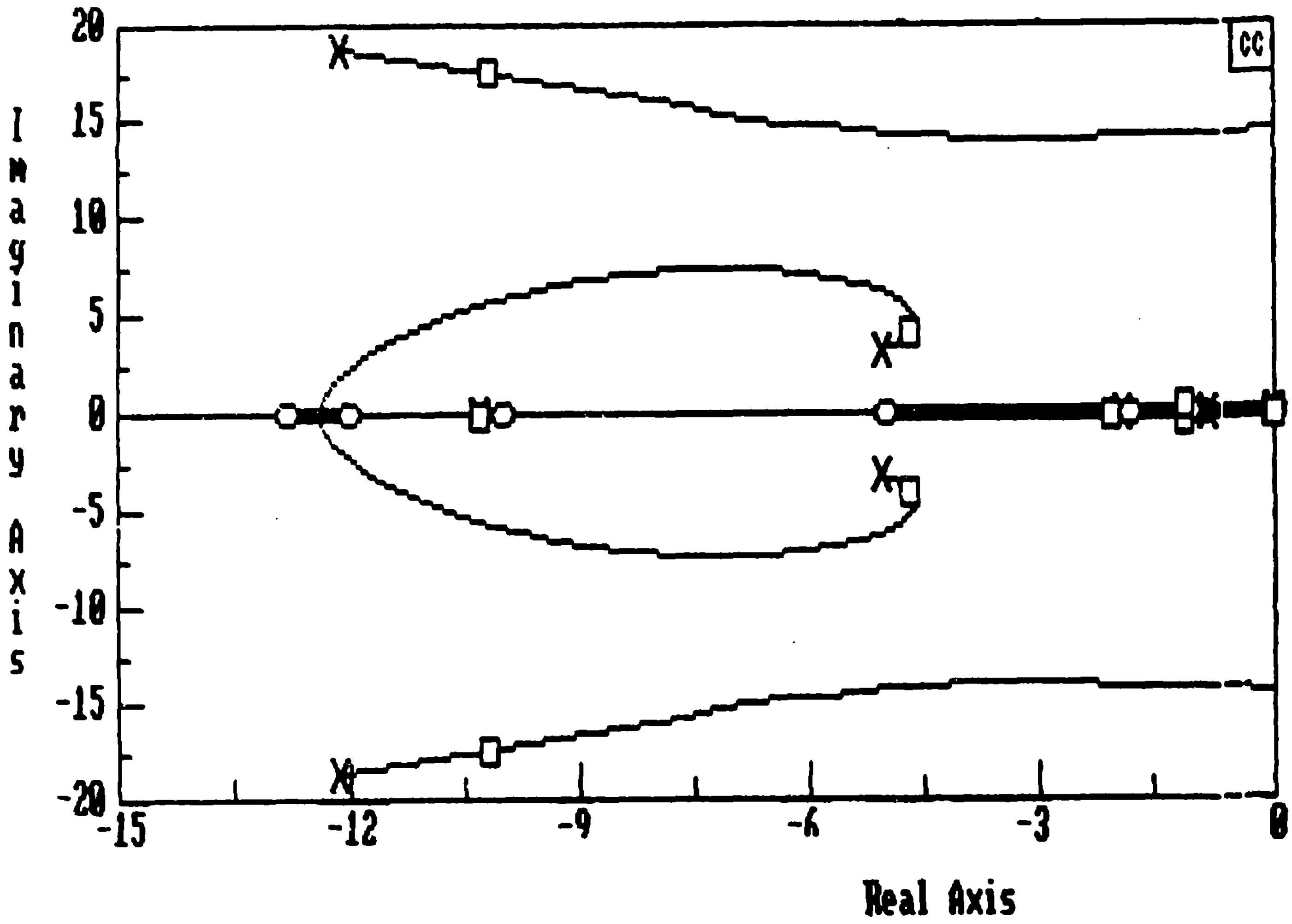


Figure F.2 Root Locus of Flight Path Angle to Flight Path Angle Command With Compensation

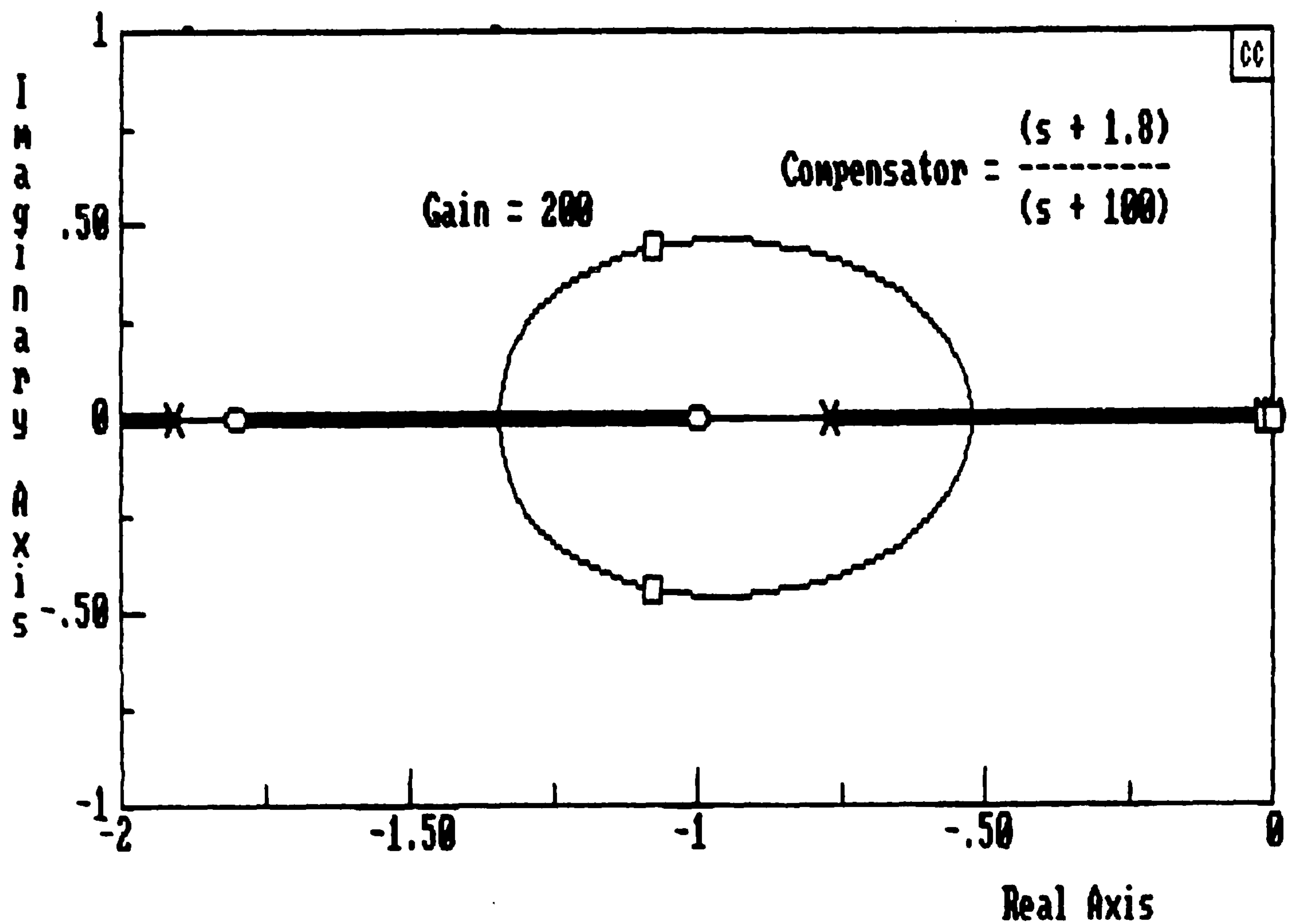


Figure F.3: Expanded View of Root Locus in Figure F.2

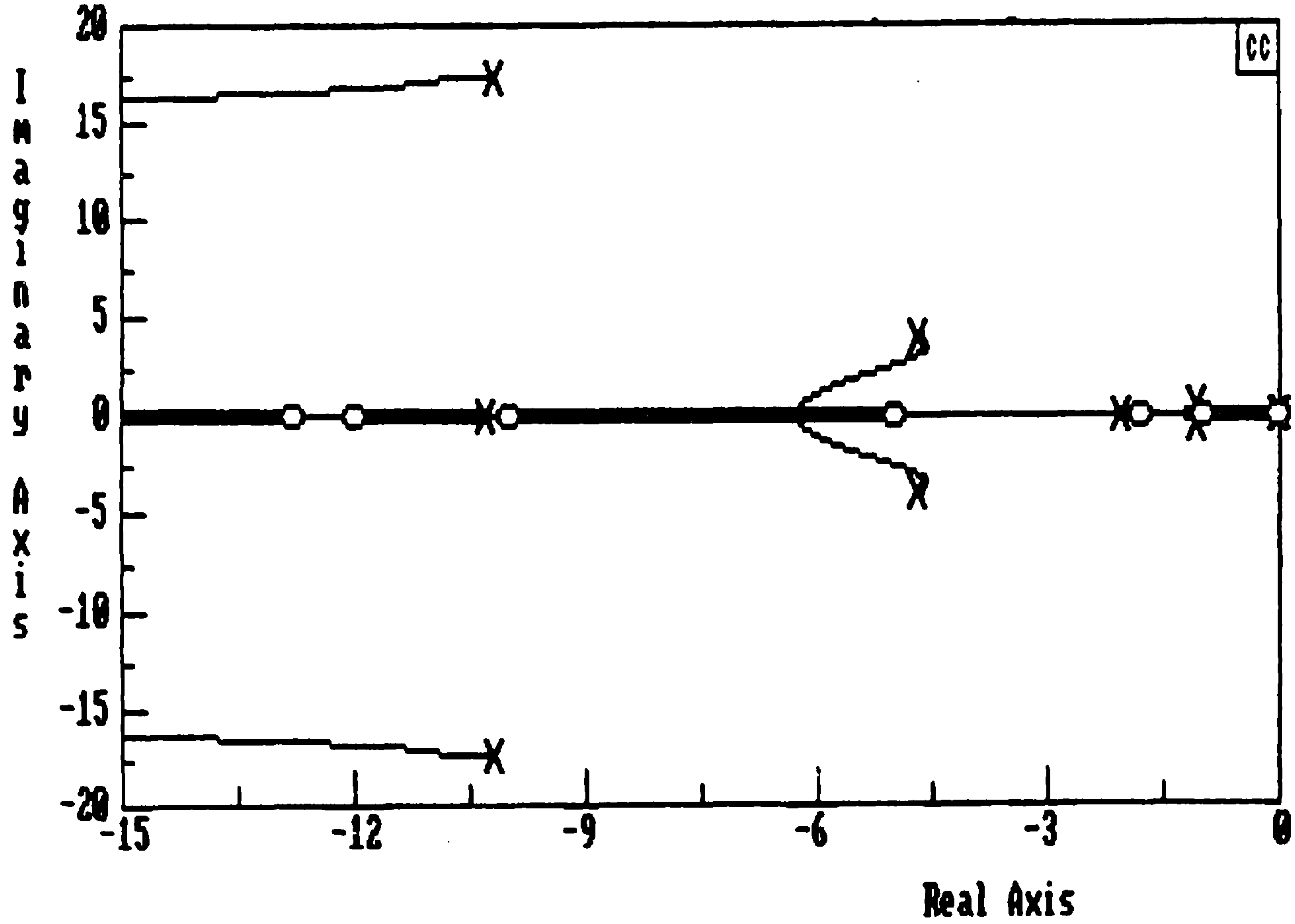


Figure F.4: Root Locus of Altitude to Flight Path Angle Command Without Compensation

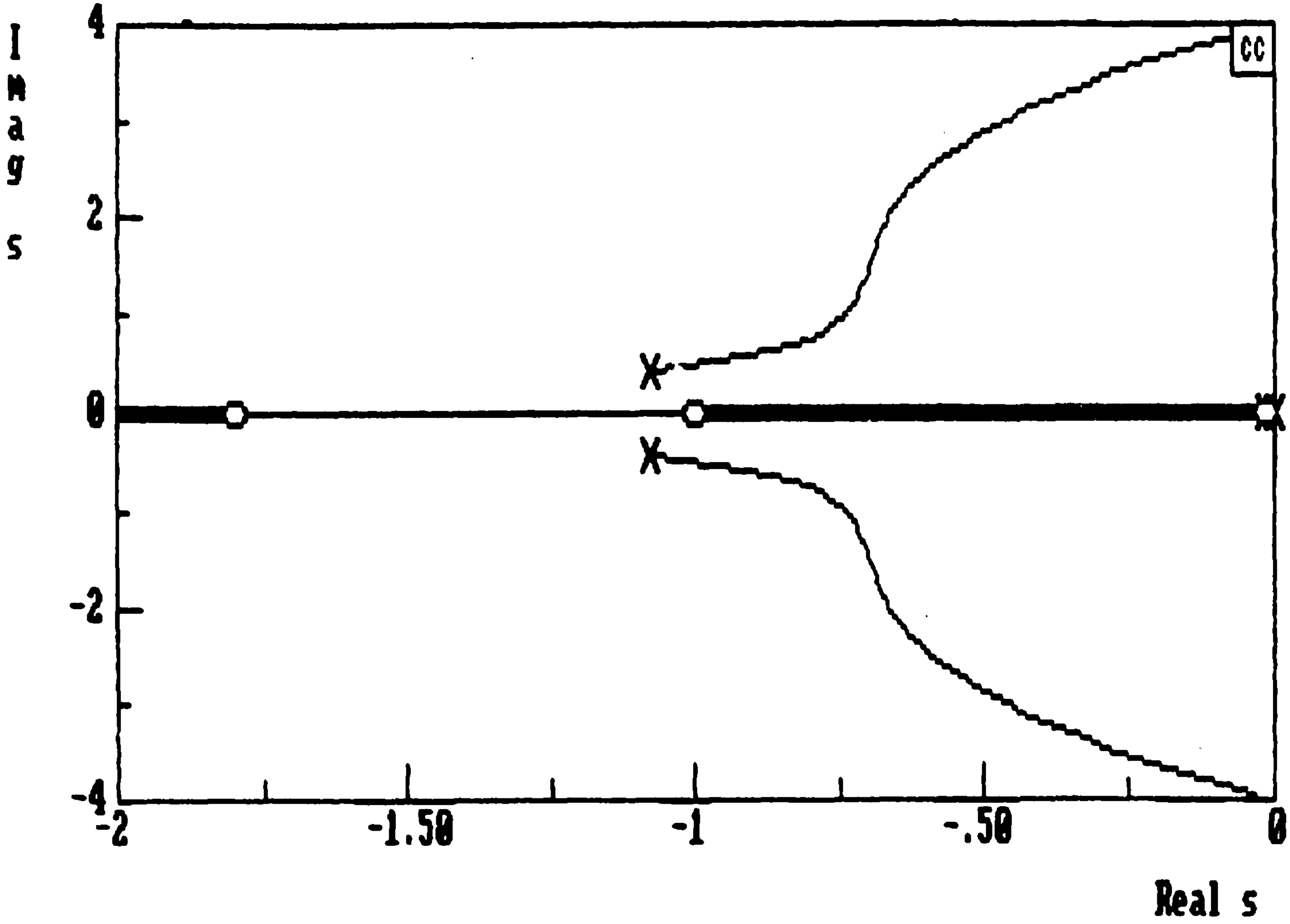


Figure F.5: Expanded View of Root Locus in Figure F.4

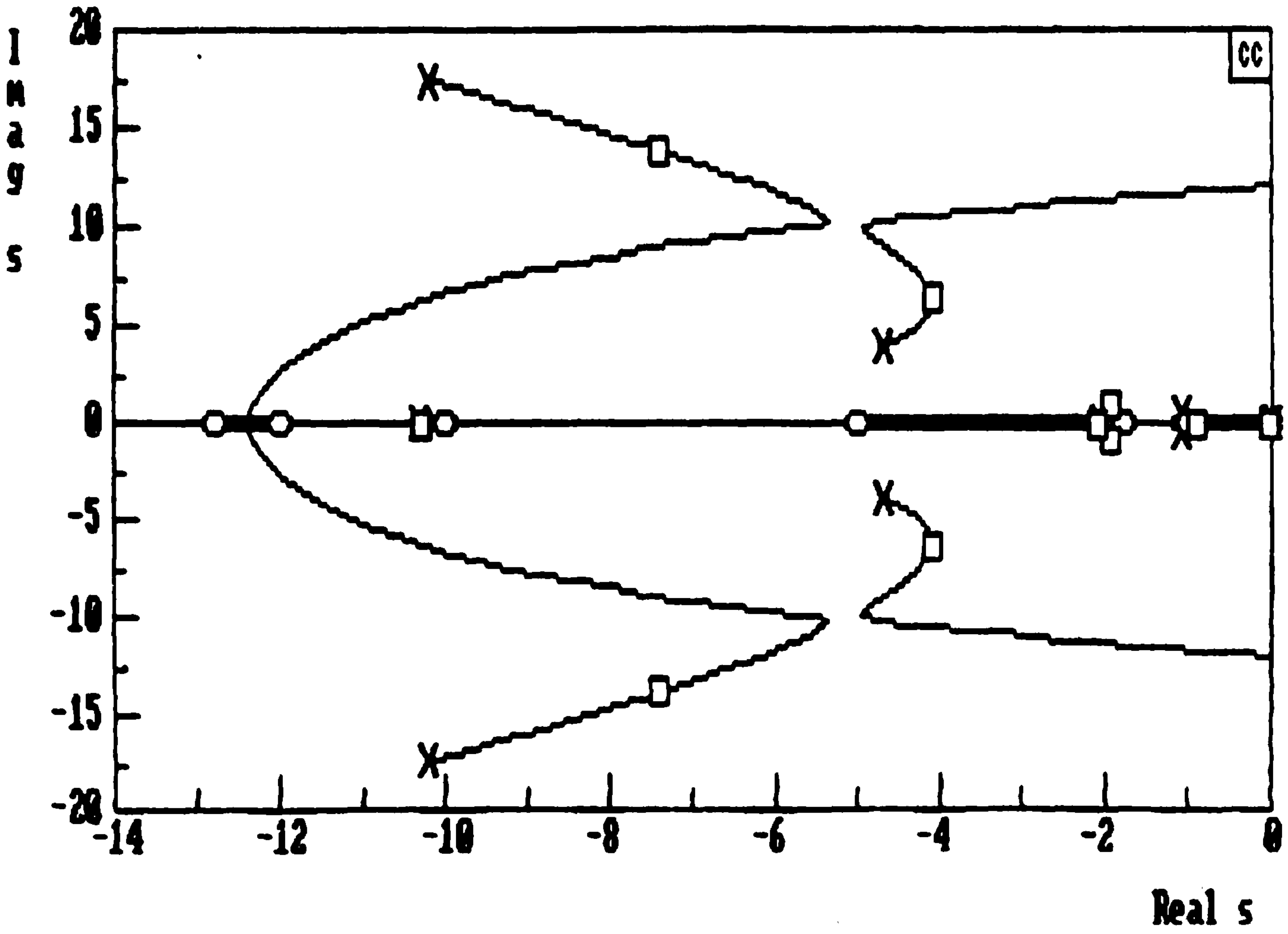


Figure F.6: Root Locus of Altitude to Flight Path Angle Command With Compensation

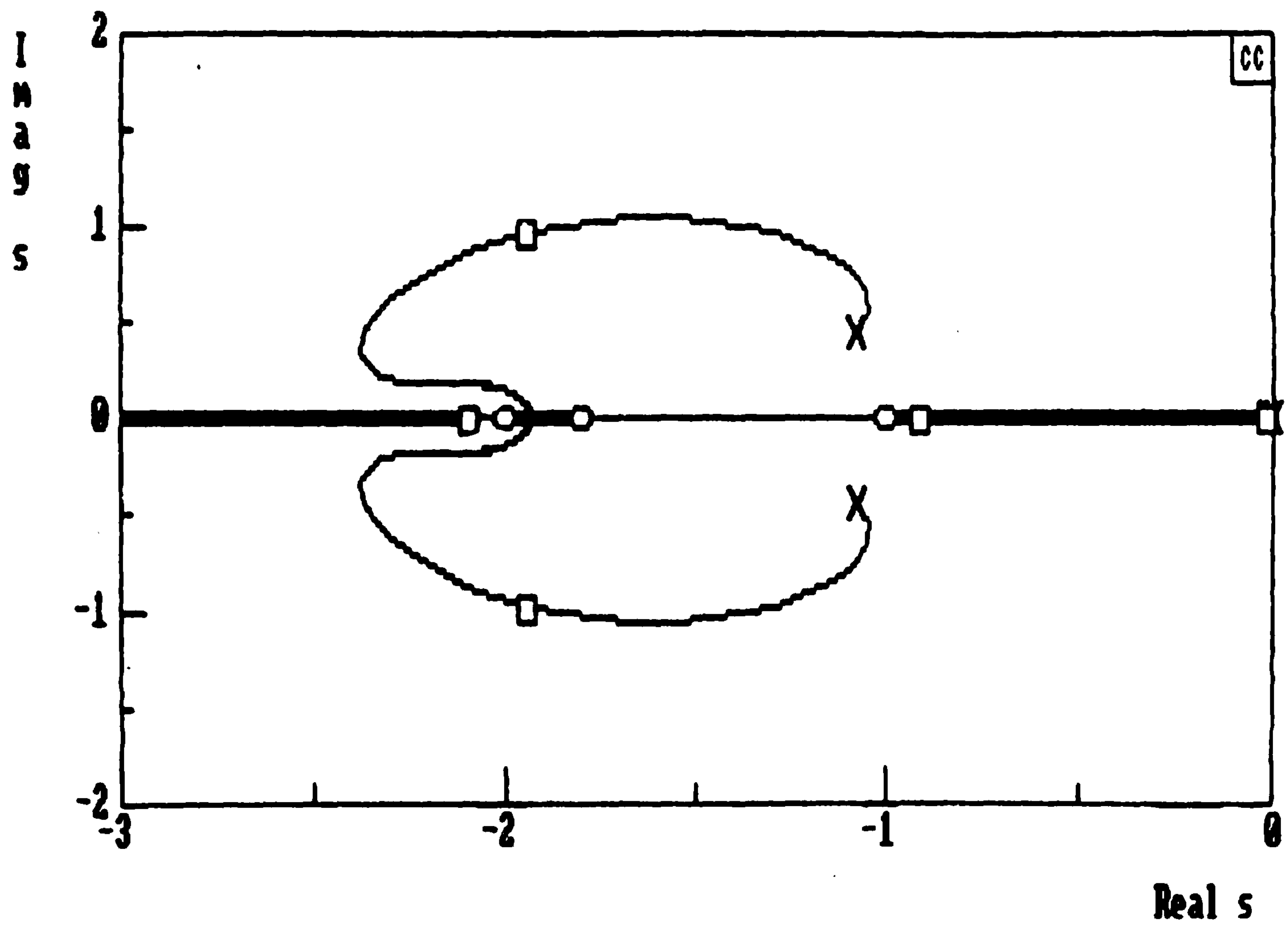


Figure F.7: Expanded View of Root Locus in Figure F.6

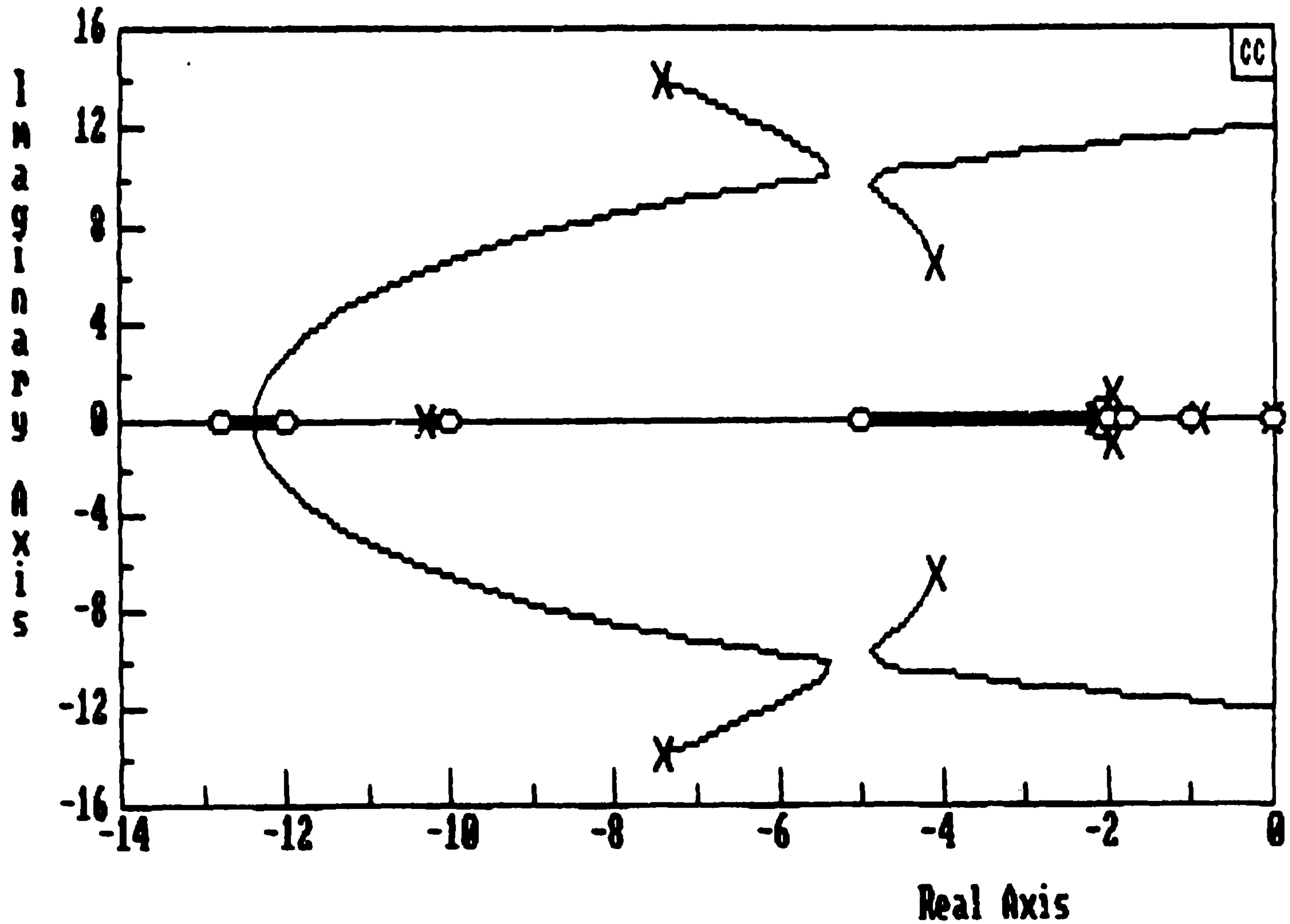


Figure F.8: Root Locus of Altitude to Altitude Command Transfer Function

Bibliography

1. Skoog, Mark A. and 1Lt. Gregory W. Bice. AFTI/F-16 AMAS Ground Collision Avoidance System Evaluation. AFFTC-TR-87-11. Edwards AFB, CA: Air Force Flight Test Center, August 1987.

2. Roskam, Jan. Airplane Flight Dynamics and Automatic Flight Controls, Part 1. Ottawa, Kansas: Roskam Aviation and Engineering Corporation, 1979.

3. McRuer, Duane et al. Aircraft Dynamics and Automatic Control. Princeton, New Jersey: Princeton University Press, 1973.

4. USAF Test Pilot School. Flying Qualities Textbook, Vol. II,Part 2. USAFTPS-CDR-86-03. Edwards AFB, CA: USAFTPS, April 1986 (AD-A170960).

5. Reid, J. Gary. Linear Systems Fundamentals. New York: Mcgraw- Hill, Inc., 1983.

6. Thompson, Peter M. Program CC Manual, Version 3.0. Systems Technology Inc., Hawthorne, California, 1985.

7. MATRIXx Program Manual, Version 7.0. Integrated Systems Inc., Santa Clara, California, October 1988.

8. Barfield, A. Finley. Multivariable Control Laws for the AFTI/F-16. MS thesis, AFIT/GE/EE/83S-4. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, July 1983.

Vita

Captain Gregory W. Bice was born on 2 September 1962 in Hays, Kansas. He graduated from high school in Beloit, Kansas, in 1981 and entered the United States Air Force Academy, receiving a Bachelor of Science degree in Aeronautical Engineering in May 1985. Upon graduation, he reported to the Air Force Flight Test Center at Edwards AFB, California, where he served as a flight dynamics engineer with the 6520th Test Group. Captain Bice was assigned to the Advanced Fighter Technology Integration (AFTI) / F-16 Joint Test Force where he worked on the development and flight testing of automated attack and ground collision avoidace systems, receiving a patent for his design of an automated ground collision avoidance system. He entered the School of Engineering, Air Force Institute of Techology, in June 1988.

Permanent address: 718 Colonial Ct.
Salina, Kansas 67401

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release; distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
AFIT/GAE/ENY/89D-03

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
School of Engineering

6b. OFFICE SYMBOL *(If applicable)*
AFIT/ENG

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS *(City, State, and ZIP Code)*
Air Force Institute of Technology
Wright-Patterson AFB OH 45433-6583

7b. ADDRESS *(City, State, and ZIP Code)*

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
Controls Applications Group

8b. OFFICE SYMBOL *(If applicable)*
WRDC/FIGX

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*
Flight Dynamics Laboratory
Wright-Patterson AFB OH 45433-6583

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*
DEVELOPMENT OF AN AUTOMATIC GROUND COLLISION AVOIDANCE SYSTEM USING A DIGITAL TERRAIN DATABASE

12. PERSONAL AUTHOR(S)
Gregory W. Bice, B.S., Capt, USAF

13a. TYPE OF REPORT
MS Thesis

13b. TIME COVERED
FROM ________ TO ______

14. DATE OF REPORT *(Year, Month, Day)*
1989 December

15. PAGE COUNT
151

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
01	02	00
01	03	12

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
F-16 Aircraft   Autopilot   Terrain Avoidance
Automatic Ground Collision Avoidance Systems (GCAS)
Controlled Flight Into Terrain (CFIT)

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Thesis Advisor: Capt Curtis P. Mracek
Associate Professor
Department of Aeronautics and Astronautics

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Curtis P. Mracek, Assoc. Professor

22b. TELEPHONE *(Include Area Code)*
(513) 255-3517

22c. OFFICE SYMBOL
ENY

DD Form 1473, JUN 86   *Previous editions are obsolete.*   SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

During the past several years, the Air Force has experienced an increasing number of single seat aircraft mishaps due to what is termed 'controlled flight into terrain'. To combat this phenomenon, several ground collision avoidance systems (GCAS) have been developed to warn the pilot of a potential collision with the terrain if some action is not taken. ~~However~~, all current systems have shortcomings pertaining to the sensors that are used and the recovery maneuver that is flown. The USAF is evaluating the potential of digital terrain databases for onboard navigation and terrain avoidance in combat aircraft. The purpose of this thesis was to develop a control system for performing terrain avoidance using a simulated terrain database. This study was conducted for an F-16 aircraft in level flight at 0.6 Mach and sea level conditions. A state-space model of the aircraft and its flight control system was developed using aircraft control derivatives, an F-16 control law diagram, and traditional linearization techniques on the aircraft equations of motion. A control system for implementing terrain avoidance was derived based on the look-ahead capability of the terrain database. Control system response was evaluated using a simulated terrain obstacle and various look-ahead distances on the terrain database. Results indicated that a 1200 foot, or roughly 1.8 second, look-ahead distance provided good improvement in terrain avoidance capabilities for the F-16 compared to looking strictly downward from the aircraft for terrain information.

END

FILMED

1-90

DTIC